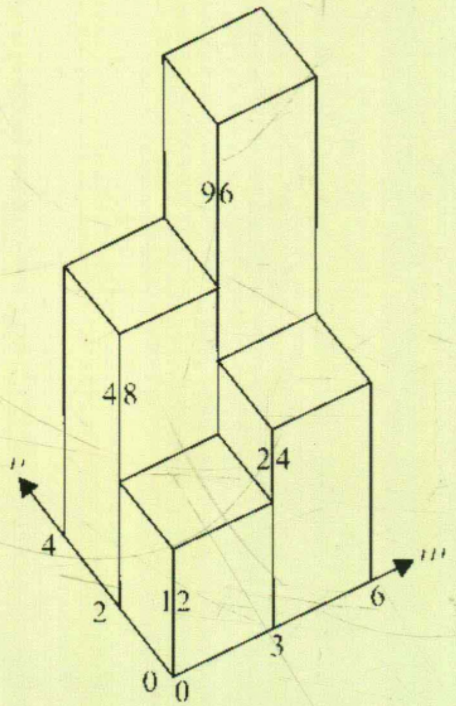
Zhou Xunwei

MUTUALLY-INVERSISTIC LOGIC, MATHEMATICS, AND THEIR APPLICATIONS



中央编译出版社
CCTP Central Compilation & Translation Press

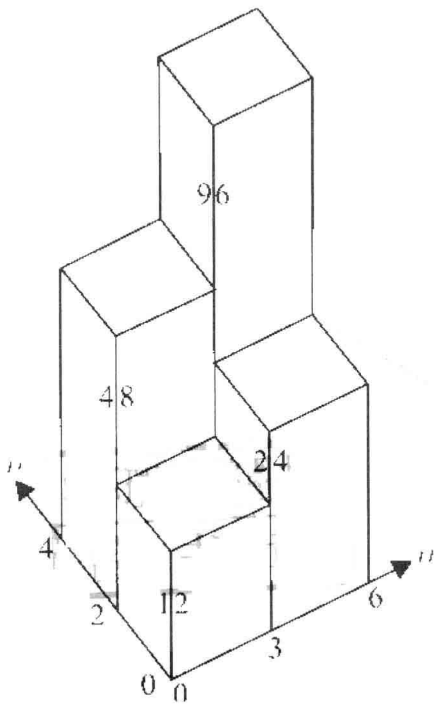
ISBN 978-7-5117-1611-8
9 787511 716118 >

责任编辑：董巍　陈振僧

定价：168.00元

Zhou Xunwei

MUTUALLY-INVERSISTIC LOGIC, MATHEMATICS, AND THEIR APPLICATIONS



CCTP 中央编译出版社
Central Compilation & Translation Press

图书在版编目(CIP)数据

互逆主义逻辑、数学和它们的应用 = Mutually-inversistic logic, mathematics, and their applications：英文 / 周训伟著.
—北京：中央编译出版社，2013.3

ISBN 978-7-5117-1611-8

Ⅰ.①互…
Ⅱ.①周…
Ⅲ.①数理逻辑-应用-英文
Ⅳ.①O142

中国版本图书馆 CIP 数据核字(2013)第 043052 号

互逆主义逻辑、数学和它们的应用(英文版)

出 版 人	刘明清
责任编辑	董　巍
责任印制	尹　珺
出版发行	中央编译出版社
地　　址	北京西城区车公庄大街乙 5 号鸿儒大厦 B 座(100044)
电　　话	(010)52612345(总编室)　(010)52612363(编辑室) (010)66161011(团购部)　(010)52612332(网络销售) (010)66130345(发行部)　(010)66509618(读者服务部)
网　　址	www.cctphome.com
经　　销	全国新华书店
印　　刷	北京瑞哲印刷厂
开　　本	787 毫米×1092 毫米　1/16
字　　数	660 千字
印　　张	31
版　　次	2013 年 3 月第 1 版第 1 次印刷
定　　价	168.00 元

Preface

In 1984, when I taught myself discrete mathematics, I found out that the definition of material implication is both valuable and defective. Its value lies in that it correctly reflects the establishment of implicational propositions. Its defect lies in that it cannot be used to make hypothetical inference, but its generalized inverse functions can make. Just like that when we know the summand 2 and the sum 5 and we want to find the addend, we cannot use addition $2+?=5$ but its inverse operation subtraction $5-2=?$ to find it.

Since then, I have been constructing mutually-inversistic mathematical logic. Now, it is fully fledged. It includes mutually-inversistic logic, mutually-inversistic mathematics, and their applications. Mutually-inversistic logic includes two calculi and four theories of mutual-inversism, mutually-inversistic granular computing, unified logics. Mutually-inversistic mathematics includes mutually-inversistic analytic geometry, mutually-inversistic mathematical analysis, mutually-inversistic abstract algebra, universal matrix. Applications include logic programming (see Part 4), automated theorem proving, planning and scheduling, database, semantic network, expert system, program verification, natural language processing, hardware verification, machine learning, data mining, data warehouse, program refinement, many-valued computer, modern control theory, etc..

All branches of mutually-inversistic mathematical logic are discrete, so, mutually-inversistic mathematical logic can also be regarded as mutually-inversistic discrete mathematics.

This monograph is suitable for faculty members, researchers, graduate students working in the field of logic, mathematics, computer science, automation to use.

My email is zhouxunwei@263.net.

CONTENTS

Part 2 Mutually-inversistic set theory

Part 5 Mutually-inversistic granular computing

Part 14 Applications of mutually-inversistic mathematics

Part 1
Mutually-inversistic logical calculus

Mutually-inversistic logical calculus is composed of mutually-inversistic propositional calculus and predicate calculus. As predicate calculus is more useful than propositional calculus, the author introduces predicate calculus first and in detail, introduces propositional calculus second and in brief.

Chapter 1
Fundaments of predicate calculus

1.1 Material implication vs. mutually inverse implication

Material implication is defined as Table 1.1.

Table 1.1 Material implication

A	B	A→B
F	F	T
F	T	T
T	F	F
T	T	T

The author finds out that Table 1.1 is both valuable and defective. The value of Table 1.1 lies in that it correctly reflects the establishment of A→B.Take "if it rains, then the ground is wet" as an example.On October 1, 2006, in Beijing, it didn't rain, and the ground was not wet. This is the first row of Table 1.1. On September 30, 2006, in Beijing, it didn't rain, but sprayers sprayed the ground wet. This is the second row of Table 1.1. On August 20, 2006, in Xi'an, it rained, and the ground was wet. This is the fourth row of Table 1.1. It is not the case that it rains and the ground is not wet. Hence, the third row of Table 1.1 never occurs. So, from "it rains" and "the ground is wet" we establish "if it rains, then the ground is wet", using Table 1.1.

Material implication has a well known defect: material implication paradox. For example, $P\wedge\neg P\rightarrow Q$ is a material implication paradox. Material implication paradox has two characteristics: (1) If the antecedent is false, or the consequent is true, then the antecedent implies the consequent; (2) there is no nexus of contents between the antecedent and the consequent. The above material implication paradox only depends on logical knowledge, the author calls it logical material implication paradox. The proposition "if snow is black, then 2+2=4"satisfies the second row of Table 1.1, it is a true proposition. It also satisfies the two characteristics of material implication paradox. The author calls it empirical or mathematical material implication paradox, because it depends on empirical or mathematical knowledge. Mutually-inversistic logic requires that the antecedent not be permanently false, the consequent not be permanently true, the antecedent and the

consequent share the same variables. In mutually-inversistic logic, the two characteristics of material implication paradox are not satisfied. So, mutually-inversistic logic is free of implication paradox.

$(P\wedge Q\rightarrow R)\rightarrow(P\rightarrow R)\vee(Q\rightarrow R)$ is a tautology. In $(P\wedge Q\rightarrow R)\rightarrow(P\rightarrow R)\vee(Q\rightarrow R)$, the antecedent is not permanently false, the consequent is not permanently true, the antecedent and the consequent share the same variables P, Q, and R. It does not satisfy the two characteristics of material implication. It is not a material implication paradox. Yet, it is still absurd. Suppose P is assigned $x\leqslant y$, Q is assigned $x\geqslant y$, R is assigned $x=y$. Then it says: if $x\leqslant y$ and $x\geqslant y$ implies $x=y$, then $x\leqslant y$ implies $x=y$, or, $x\geqslant y$ implies $x=y$. Absurd! Its corresponding proposition in mutually-inversistic logic is not a logical theorem.

The author finds out Table 1.1 has a less obvious, but more serious defect: it cannot be used to make hypothetical inference. Proponents of classical logic say: "Table 1.1 can be used to make hypothetical inference. Take the affirmative expression of hypothetical inference as an example, both A→B and A being true are the fourth row of Table 1.1, in which B is true. Thus, from A→B being true and A being true we infer B being true."But there is a principle in philosophy: human cognition is from the known to the unknown. Table 1.1 is a truth function. And there is a principle in mathematics: an evaluation of a function is from its arguments to its value. If we want to evaluate from its value to its argument, then we should use its inverse functions. In order to mathematize human cognition, we let the known be the arguments, let the unknown be the value, so that the human cognition from the known to the unknown becomes the evaluation of the function from the arguments to the value. In Table 1.1, A and B are the known, the arguments, A→B is the unknown, the value, therefore, Table 1.1 can only be used to establish A→B from A and B. Using Table 1.1 to make hypothetical inference is from the unknown to the known, from the value to the argument, is in violation of the principles of philosophy and mathematics.

Although Table 1.1 cannot be used to make hypothetical inference, in its inverse functions, A or B is a value, A→B is an argument, so, its inverse functions can be used to make hypothetical inference. Following this clue, mutually inverse implication is defined as Tables 1.2 to 1.4.

Table 1.2
Inductive composition for $\leqslant^{-1}$

tv(A)	tv(B)	tv(A $\leqslant^{-1}$B)
F	F	T
F	T	n
T	F	F
T	T	T

Table 1.3
Decomposition one for $\leqslant^{-1}$

tv(A $\leqslant^{-1}$B)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	u
T	T	T

Table 1.4
Decomposition two for $\leqslant^{-1}$

tv(A $\leqslant^{-1}$B)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	F
T	T	u

Tv in Tables 1.2 to 1.4 means truth value. N in Table 1.2 means "need not determine". U in Tables 1.3 and 1.4 means "unable to determine".Table 1.2 is similar to Table 1.1 except for the second row (we will explain later why the second row is dissimilar), it is used to establish $A\leqslant^{-1}B$ from A and B. Tables 1.3 and 1.4 are the generalized inverse functions of Table 1.2, $A\leqslant^{-1}B$ in them are the known, the argument. After $A\leqslant^{-1}B$ is established in Table 1.2, it becomes known, becomes the argument, then we can use Tables 1.3 and 1.4 to make hypothetical inference. Table 1.3 is the affirmative expression of hypothetical inference: when $A\leqslant^{-1}B$ is true and A is true, then we can infer that B is true; in the other three cases, we cannot determine the truth value of B. Table 1.4 is the negative expression of hypothetical inference: when $A\leqslant^{-1}B$ is true and B is false, we can infer that A is false; in the other three cases, we cannot determine the truth value of A. In Tables 1.2 to 1.4, A and B are the special, $A\leqslant^{-1}B$ is the general. Table 1.2 is from the special to the general, it is called inductive composition. Tables 1.3 and 1.4 are from the general to the special, they are called decomposition. Inductive composition and decomposition are mutually inverse human cognitive processes, hence mutually inverse implication gets its name.

1.2 Formation of terms and propositions

1.2.1 Logical operators

The Venn diagrams of the connectives in classical logic are shown Fig. 1.1.

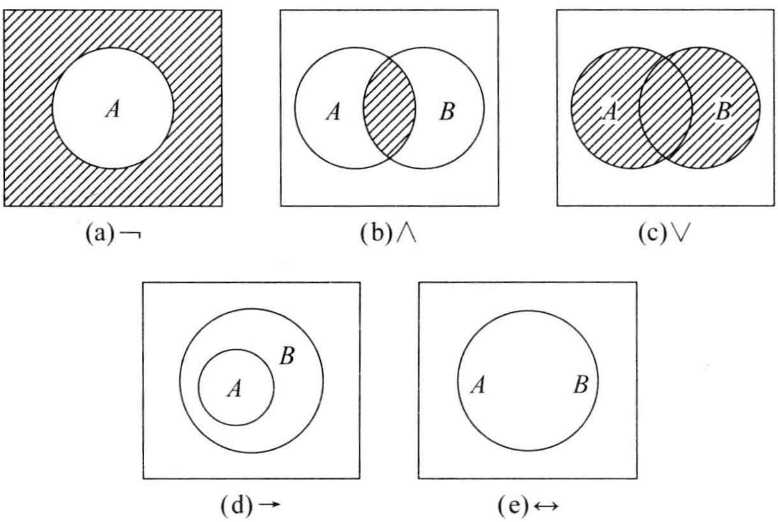


Fig. 1.1 The Venn diagrams of the connectives in classical logic

From Fig. 1.1 (a), (b), and(c) we see that $\neg$, $\wedge$ and $\vee$ are from set(s) to shadowed sets. In mutually-inversistic logic, they are called composition operators. From Fig. 1.1 (d) and

(e) we see that $\rightarrow$ and $\leftrightarrow$ represent connections between two sets, they cannot be denoted by shadowed sets, but by relative positions between two sets. In mutually-inversistic logic, they are called connection operators.

There are 12 logical operators in mutually-inversistic logic. Four of them are composition operators, they are, according to priority, $\neg$ (negation), $\wedge$(conjunction), $\vee$((compatible) disjunction), $\oplus$(incompatible disjunction). Eight of them are connection operators, they are, according to priority, $/\wedge^{-1}$(mutually inverse conjunction), $\vee/^{-1}$(mutually inverse disjunction), $\leqslant^{-1}$(mutually inverse implication, sufficient condition), $=^{-1}$(mutually inverse equivalence, sufficient and necessary condition), $<^{-1}$(mutually inverse proper implication, sufficient but not necessary condition), $\vee^{-1}$(mutually inverse compatible disjunction), $\oplus^{-1}$(mutually inverse incompatible disjunction), $\times^{-1}$(mutually inverse intercross). Connection operators are divided into unicellular connection operators and multi-cellular connection operators. $=^{-1}$, $<^{-1}$, $\vee^{-1}$, $\oplus^{-1}$, and $\times^{-1}$ are unicellular connection operators. $/\wedge^{-1}$, $\vee/^{-1}$, and $\leqslant^{-1}$ are multi-cellular connection operators.

1.2.2 Knowledge constituents

Knowledge is composed of knowledge constituents. Knowledge constituents in predicate calculus include term constants and variables, function constants and variables, predicate constants and variables, fact composition operator constants and variables, empirical or mathematical connection operator constants and variables, empirical or mathematical composition operator constants and variables, logic connection operator constants and variables. The classification of knowledge constituents is shown in Fig. 1.2.

logical connection operator constants $\leqslant^{-1}$, $=^{-1}$	logical connection operator variables λ, δ, ξ	empirical or mathematical composition operator constants $\neg$, $\wedge$, $\vee$	empirical or mathematical composition operator variables η	fourth-order constituent
empirical or mathematical connection operator constants $\leqslant^{-1}$, $/\wedge^{-1}$	empirical or mathematical connection operator variables ψ, ω, θ	fact composition operator constants $\neg$, $\wedge$, $\vee$	fact composition operator variables μ	third-order constituent
predicate constants $<$, $=$, $\leqslant$	predicate variables p, q, r	function constants $+$, $-$, $*$, $\div$	function variables f, g, h	second-order constituent
term constants 0, 1, 2	term variables x, y, z			first-order constituent
main constants	main variables	auxiliary constants	auxiliary variables	

the main   the auxiliary

Fig. 1.2 The classification of knowledge constituents

From Fig. 1.2 we learn that, horizontally, terms are first-order constituents, predicates and functions are second-order constituents, empirical or mathematical connection operators and fact composition operators are third-order constituents, logical connection operators and empirical or mathematical composition operators are fourth-order constituents; vertically, terms, predicates, and connection operators are the main, functions and composition operators are the auxiliary. The auxiliary connecting knowledge forms the same grade knowledge. The main (except terms) connecting knowledge forms knowledge one grade higher.

1.2.3 Propositions and mutually-inversistic sets are identical

1.2.3.1 Propositions and their truth value

There are propositions in logic. For example, "the Earth turns around the Sun" and "the Sun turns around the Earth" are propositions. There are true propositions and false ones. "The Earth turns around the Sun" is a true proposition while "the Sun turns around the Earth" is a false one. The truth and falsehood of a proposition is denoted by its truth value. Truth is denoted by "T", falsehood by "F". Tv is a truth mapping function, mapping a proposition to its truth value. Suppose A is a proposition, tv(A) is its truth value, then we use binary tuple $\langle$A, tv(A)$\rangle$ to describe A. For example, "the Earth turns around the Sun" being true can be written as:

$\langle$turn_around(Earth, Sun), T$\rangle$. (1.1)

"The Sun turns around the Earth" being false can be written as:

$\langle$turn_around(Sun, Earth), F$\rangle$. (1.2)

Only by combining a proposition with its truth value can we fully describe a proposition.

Some propositions such as (1.1) and (1.2) have definite truth value. They are called invariable propositions, corresponding to propositions in classical logic. The truth value of some proposition varies with different assignments to its variable. Such propositions are called variable propositions, corresponding to propositional functions in classical logic. For example,

turn_around (x, y) (1.3)

is a variable proposition. When x is assigned Earth, y is Sun, it is true; when x is assigned Sun, y Earth, it is false.

1.2.3.2 Mutually-inversistic sets

A set is composed of elements. Suppose element e_1 is a member of the set S, then we say "e_1 belongs to S", denoted by $e_1 \in S$. Suppose e_2 is not a member of the set S, then we say "e_2 doesn't belong to S", denoted by $e_2 \notin S$. One way of describing a set is by listing its elements between braces. Suppose set S is composed of e_1 and e_2, then we write $S=\{e_1, e_2\}$.

The other way of describing a set is by describing the property its elements satisfies, e.g. S={x|x is a natural number}. If every element of S_1 is also an element of S_2, then S_1 is a subset of S_2. The universe of discourse is called the universal set, denoted by U. Any set is a subset of the universal set. A set that has no element in it is called the empty set, denoted by $\varnothing$.

Mutually-inversistic sets are obtained by extending the concept set in three ways. First, the author deems that a set can be composed not only of elements, but also of n-tuples, which is a sequence made up of n elements (e_1, e_2, … , e_n). All of the n-tuples comprises an n-dimensional space, which is also called the universal set, denoted by U. The universe of discourse is the one-dimensional space. A subset of the n-dimensional space is a set, which is composed of n-tuples. A set that has no n-tuple in it is called the empty set, denoted by $\varnothing$. Secondly, the author deems that a set is divided into zeroth-level set, first-level set, and second-level set, and an element is divided into zeroth-level element, first-level element, and second-level element, accordingly. Thirdly, the author deems that a set is divided into set constant and set variable, and an element is divided into element constant and element variable, accordingly.

1.2.3.3 Propositions and mutually-inversistic sets are identical

A variable proposition does not have a definite truth value; however, sometimes we associate it with a truth value in order to denote a set. For example, <turn_around(x, y), T> denotes the set {(Earth, Sun), (Moon, Earth), …}. <turn_around(x, y), F> denotes the set {(Sun, Earth),(Earth, Moon), …}. <turn_around(x, y), F> is equivalent to <$\neg$turn_around(x, y), T>. The truth values of <turn_around(x, y), T> and <$\neg$turn_around(x, y), T> are T, so we often omit T to obtain turn_around(x, y) and $\neg$turn_around(x, y) which are not only variable propositions but also sets.

The invariable proposition <turn_around(Earth, Sun), T> can also be regarded as having the binary tuple (Earth, Sun) belonging to the set turn-around(x, y), denoted by:

$$\text{(Earth, Sun)} \in \text{turn\_around}(x, y). \tag{1.4}$$

Likewise, <turn_around(Sun, Earth), F> by:

$$\text{(Sun, Earth)} \notin \text{turn\_around}(x, y). \tag{1.5}$$

Comparing (1.1) with (1.4), (1.2) with (1.5), we see that they are two different ways of expressing the same thing; therefore, propositions and mutually-inversistic sets are identical.

1.2.4 Formation of terms

The numbers 0, 1, 2, 3, …, and the proper nouns Aristotle, Russell, Sun, Earth, etc., are, from the knowledge constituent perspective, term constants, first-order main constants; from the knowledge perspective, zeroth-order terms; corresponding to individual constants in

classical logic. 0, 1, and ∞ are distinguished terms. X, y, z are, from the knowledge constituent perspective, term variables, first-order main variables; from the knowledge perspective, first-order terms; corresponding to individual variables in classical logic, ranging over term constants. Zeroth-order term and first-order term are designated by a joint name, terms.

$+$, $-$, $*$, $\div$, etc., and height, weight, etc., are function constants or second-order auxiliary constants. A function constant maps n terms to a term. For example, $*$ in $2*3=6$ maps 2 and 3 to 6. F, g, h are function variables or second-order auxiliary variables ranging over function constants. Function constant and function variable are designated by a joint name, functions. A function connecting terms still forms a term, e.g., 2+3, height (Aristotle), $f(x, y)$ are still terms.

1.2.5 Formation of fact propositions and quasi-empirical or mathematical connection propositions

Man, mortal, integer, rational, etc., and $<$, $=$, $\leqslant$, etc., are predicate constants or second-order main constants. P, q, r are predicate variables or second-order main variables ranging over predicate constants. Predicate constant and predicate variable are designated by a joint name, predicates.

A predicate constant connecting term constants forms a zeroth-order fact proposition, corresponding to the filling of a predicate in classical logic. For example, man (Aristotle), turn_around (Earth, Sun) are zeroth-order fact propositions. A zeroth-order fact proposition is an invariable proposition.

A predicate constant connecting term variables forms a first-order fact proposition, if the proposition formed is a variable proposition. For example, man(x), turn_around(x, y) are first-order fact propositions corresponding to the naming of a predicate in classical logic.

Predicate constants $<$, $=$, $\leqslant$, etc., connecting term variables or distinguished terms 0 and 1 form quasi-empirical or mathematical connection propositions, if the proposition formed is an invariable proposition; e.g., $x=x$, $x+(-x)=0$, $x*0=0$ are quasi-empirical or mathematical connection propositions. A true quasi-empirical or mathematical connection proposition is a quasi-empirical or mathematical theorem.

A predicate variable connecting distinct term variables forms a second-order fact proposition, which is a variable proposition; e.g., $p(x)$ and $q(x, y)$ are second-order fact propositions.

P, Q, R are fact propositional variables, and are abbreviations for second-order fact propositions ranging over first-order fact propositions; e.g., P is an abbreviation for $p(x)$, $p(x, y)$, and $p(x, y, z)$.

Fact propositions composed by a composition operator still form a fact proposition. The composition operator is called a fact composition operator or a third-order auxiliary

constant. For example, $\neg$ in $\neg$integer (13.6) and $\wedge$ in parent(x, y) $\wedge$ ancestor(y, z) are fact composition operators. $\neg$integer (13.6), parent(x, y) $\wedge$ ancestor(y, z), and $P \wedge Q$ are still fact propositions. The lower case Greek letter μ denotes a fact composition operator variable or third-order auxiliary variable ranging over fact composition operators $\neg$, $\wedge$, $\vee$, $\oplus$.

$\varnothing_0$ is a permanently false fact proposition, U_0 is a permanently true fact proposition. $\varnothing_0$ and U_0 are distinguished fact propositions.

1.2.6 Formation of single empirical or mathematical connection propositions, quasi-logical connection propositions, and multiple empirical or mathematical connection propositions

Two zeroth-order fact propositions connected by a connection operator form a zeroth-order single empirical or mathematical connection proposition, which is an invariable proposition. For example, man (Aristotle) $\leqslant^{-1}$mortal (Aristotle) is a zeroth-order single empirical or mathematical connection proposition.

Two first-order fact propositions connected by a connection operator form a first-order single empirical or mathematical connection proposition, which is an invariable proposition. For example, man(x) $\leqslant^{-1}$ mortal(x) (corresponding to $\forall$x(man(x)$\rightarrow$mortal(x)) in classical logic), even_number(x) $/\wedge^{-1}$ prime_number(x) (corresponding to $\exists$x(even_number(x) $\wedge$ prime_number(x)) in classical logic), parent(x, y) $\wedge$ancestor(y, z) $\leqslant^{-1}$ancestor(x, z) are first-order single empirical or mathematical connection propositions. A true single empirical or mathematical connection proposition is a single empirical or mathematical theorem.

Two second-order fact propositions connected by a connection operator form a second-order single empirical or mathematical connection proposition, if the proposition formed is a variable proposition. For example, $P \leqslant^{-1} Q$, $P /\wedge^{-1} Q$ are second-order single empirical or mathematical connection propositions.

Two second-order fact propositions or distinguished fact propositions $\varnothing_0$, U_0 connected by$=^{-1}$ form a quasi-logical connection proposition, if the proposition formed is an invariable proposition; e.g., $P =^{-1} P$, $P \wedge \neg P =^{-1} \varnothing_0$, $P \wedge \varnothing_0 =^{-1} \varnothing_0$ are quasi-logical connection propositions. Two fact propositions connected by$\leqslant^{-1}$form a quasi-logical connection proposition, if the proposition formed is an invariable proposition; e.g., $P \leqslant^{-1} P$, $P \wedge Q \leqslant^{-1} P$, $P \leqslant^{-1} P \vee Q$ are quasi-logical connection propositions. A true quasi-logical connection proposition is a quasi-logical theorem.

The connection operator connecting fact propositions is called an empirical or mathematical connection operator or a third-order main constant. The lower case Greek letters ψ, ω, θ denote empirical or mathematical connection operator variables or third-order main variables ranging over empirical or mathematical connection operators $/\wedge^{-1}$, $\vee/^{-1}$, $\leqslant^{-1}$, $=^{-1}$, $<^{-1}$, $\vee^{-1}$, $\oplus^{-1}$, and $\times^{-1}$.

An empirical or mathematical connection operator variable connecting two distinct second-order fact propositions forms a third-order single empirical or mathematical connection proposition, which is a variable proposition; e.g., $P\psi Q$ is a third-order single empirical or mathematical connection proposition.

The upper case Greek letters Ψ, Ω, Θ are single empirical or mathematical connection proposition variables, and are abbreviations for third-order single empirical or mathematical connection propositions ranging over second-order single empirical or mathematical connection propositions. For example, Ψ is an abbreviation for $P\psi Q$.

Single empirical or mathematical connection propositions composed by a composition operator still form a single empirical or mathematical connection proposition. The composition operator is called an empirical or mathematical composition operator. For example, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ is still a single empirical or mathematical connection proposition, $\wedge$ in $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ is an empirical or mathematical composition operator. The lower case Greek letter η denotes an empirical or mathematical composition operator variable or a fourth-order auxiliary variable ranging over empirical or mathematical composition operators $\neg$, $\wedge$, $\vee$, $\oplus$.

$\varnothing_1$ is a permanently false single empirical or mathematical connection proposition, U_1 is a permanently true single empirical or mathematical connection proposition. $\varnothing_1$ and U_1 are distinguished single empirical or mathematical connection propositions.

Three or more fact propositions connected by two or more empirical or mathematical connection operators $\leqslant^{-1}$ or $/\wedge^{-1}$ form a multiple empirical or mathematical connection proposition. For example, natural_number(x)$/\wedge^{-1}${natural_number(y) $\leqslant^{-1}x\leqslant y$} (there is the least natural number) is a first-order multiple empirical or mathematical connection proposition, an invariable proposition. A true first-order multiple empirical or mathematical connection proposition is called a multiple empirical or mathematical theorem. $P(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}$ is a second-order multiple empirical or mathematical connection proposition, a variable proposition.

1.2.7 Formation of single logical connection propositions, quasi-transcendent logical connection propositions, semilogical connection propositions, and multiple logical connection propositions

Two zeroth-order single empirical or mathematical connection propositions connected by a connection operator form a zeroth-order single logical connection proposition, which is an invariable proposition. For example, {man(Aristotle) $\leqslant^{-1}$mortal(Aristotle)}$=^{-1}$ {$\neg$mortal(Aristotle) $\leqslant^{-1}\neg$man(Aristotle)} is a zeroth-order single logical connection proposition.

Two first-order single empirical or mathematical connection propositions connected by a connection operator form a first-order single logical connection proposition, which is an invariable proposition. For example, $\{\text{man}(x) \leqslant^{-1} \text{mortal}(x)\} =^{-1} \{\neg\text{mortal}(x) \leqslant^{-1} \neg\text{man}(x)\}$ is a first-order single logical connection proposition.

Two second-order single empirical or mathematical connection propositions connected by a connection operator form a second-order single logical connection proposition, which is an invariable proposition. For example, $\{P \leqslant^{-1} Q\} =^{-1} \{\neg Q \leqslant^{-1} \neg P\}$, $\{P \leqslant^{-1} Q\} \leqslant^{-1} \{P / \wedge^{-1} Q\}$, $\{P \leqslant^{-1} Q\} \wedge \{Q \leqslant^{-1} R\} \leqslant^{-1} \{P \leqslant^{-1} R\}$ are second-order single logical connection propositions. A true second-order single logical connection proposition is a single logical theorem.

Two third-order single empirical or mathematical connection propositions connected by a connection operator form a third-order single logical connection proposition, if the proposition formed is a variable proposition; e.g., $\Psi \leqslant^{-1} \Omega$ is a third-order single logical connection proposition.

Two third-order single empirical or mathematical connection propositions or distinguished single empirical or mathematical connection propositions $\varnothing_1$ and U_1 connected by $=^{-1}$ form a quasi-transcendent logical connection proposition, if the proposition formed is an invariable proposition. For example, $\Psi =^{-1} \Psi$, $\Psi \vee \neg \Psi =^{-1} \mathrm{U}_1$, are quasi-transcendent logical connection propositions. Two third-order single empirical or mathematical connection propositions connected by $\leqslant^{-1}$ form a quasi-transcendent logical connection proposition, if the proposition formed is an invariable proposition; e.g., $\Psi \leqslant^{-1} \Psi$, $\Psi \wedge \Omega \leqslant^{-1} \Psi$ are quasi-transcendent logical connection propositions. A true quasi-transcendent logical connection proposition is a quasi-transcendent logical theorem.

The connection operator connecting single empirical or mathematical connection propositions is called a logical connection operator or a fourth-order main constant. The lower case Greek letters λ, δ, ξ denote logical connection operator variables or fourth-order main variables ranging over logical connection operators $\vee/^{-1}$, $\leqslant^{-1}$, $=^{-1}$, $<^{-1}$, $\vee^{-1}$, and $\oplus^{-1}$.

A logical connection operator variable connecting two distinct third-order single empirical or mathematical connection propositions forms a fourth-order single logical connection proposition, which is a variable proposition; e.g., $\Psi \lambda \Omega$ is a fourth-order single logical connection proposition.

The upper case Greek letters Λ, Δ, Ξ are single logical connection proposition variables, and are abbreviations for fourth-order single logical connection propositions ranging over third-order single logical connection propositions; e.g., Λ is an abbreviation of $\Psi \lambda \Omega$.

$\varnothing_2$ is a permanently false single logical connection proposition, U_2 is a permanently true single logical connection proposition. $\varnothing_2$ and U_2 are distinguished single logical connection propositions.

If on one side of the connection operator is a single empirical or mathematical connec-

tion proposition, on the other side of the connection operator is a fact proposition, then the proposition formed is a semilogical connection proposition; e.g., $\{\neg P <^{-1} \neg Q\} \wedge \{\neg P <^{-1} Q\} =^{-1} P$ is a semilogical connection proposition.

Two multiple empirical or mathematical connection propositions connected by a logical connection operator form a multiple logical connection proposition. For example, $\{\{\text{natural\_number}(x) / \wedge^{-1} \{\text{natural\_number}(y) \leqslant^{-1} x \leqslant y\}\} \leqslant^{-1} \{\text{natural\_number}(y) \leqslant^{-1} \text{natural\_number}(x) / \wedge^{-1} x \leqslant y\}$ is a first-order multiple logical connection proposition, an invariable proposition. $\{\{p(x) / \wedge^{-1} \{q(y) \leqslant^{-1} r(x, y)\}\} \leqslant^{-1} \{q(y) \leqslant^{-1} p(x) / \wedge^{-1} r(x, y)\}$ is a second-order multiple logical connection proposition, an invariable proposition. A true second-order multiple logical connection proposition is a multiple logical theorem.

The order of a term or a proposition is specified according to the highest order of main variables appearing in it. For example, $p(x, y)$ is a second-order fact proposition, because the highest order of main variables in it is the second-order main variable p.

1.2.8 Summary of terms and propositions

The terms and propositions described in Sections 1.2.4 to 1.2.7 are knowledge, the formation of which is shown in Fig. 1.3.

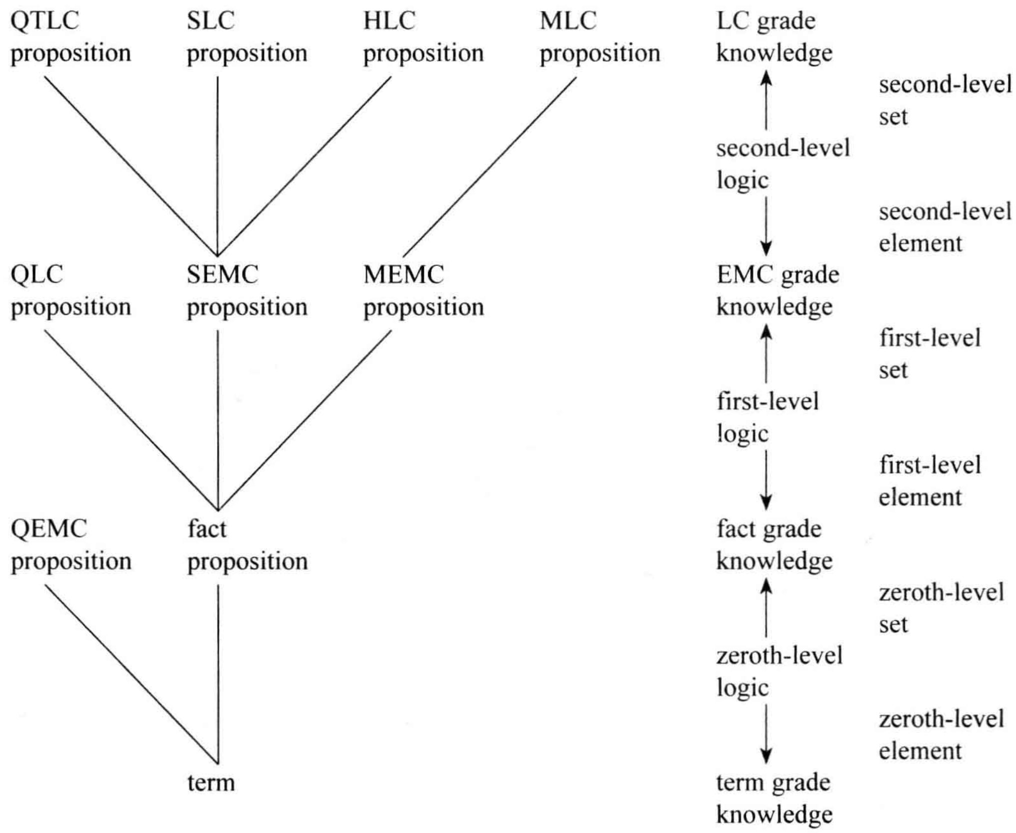


Fig. 1.3 Formation of knowledge

MEMC proposition in Fig. 1.3 stands for multiple empirical or mathematical connection proposition, the rest can be inferred by analogy. From Fig. 1.3 we see that knowledge is divided into four grades: term grade, fact grade, empirical or mathematical connection grade, and logical connection grade. From the formation point of view, a quasi-empirical or mathematical connection proposition is similar to a fact proposition, both are formed by a predicate connecting terms. From the point of view of boundness of variables, a quasi-empirical or mathematical connection proposition is similar to a single empirical or mathematical connection proposition, both are bound on term variables. In Fig. 1.3, the quasi-empirical or mathematical connection proposition is classified as fact grade knowledge. A similar discussion can be held for quasi-logical connection propositions and quasi-transcendent logical connection propositions. In Fig. 1.3, the relationship among the four grades of knowledge, three levels of logic, three levels of elements, and sets is also given.

Term—fact proposition—single empirical or mathematical connection proposition—single logical connection proposition is the main thread in Fig. 1.3, their further classification is shown in Fig. 1.4.

Invariable proposition

0SLC proposition	1SLC proposition	2SLC proposition	3SLC proposition	4SLC proposition	SLC proposition
0SEMC proposition	1SEMC proposition	2SEMC proposition	3SEMC proposition		SEMC proposition
0F proposition	1F proposition	2F proposition			Fact proposition
0 term	1 term				term
zeroth-order	first-order	second-order	third-order	fourth-order	variable proposition

Fig. 1.4 Further classification of the main thread

2SLC proposition in Fig. 1.4 stands for second-order single logical connection proposition, the rest can be inferred by analogy. From Fig. 1.4 we see that the main thread is horizontally classified into four grades—terms, fact propositions, single empirical or mathematical connection propositions, single logical connection propositions—and is vertically classified into zeroth-order, first-order, second-order, third-order, fourth-order knowledge. The propositions in Fig. 1.4 are divided into invariable propositions and variable propositions. On the upper left side of the double line are invariable propositions; on the lower right side, variable propositions.

1.2.9 Instance

When the highest order variables in a proposition are assigned to the same order constants, the proposition is said to be instancized: the proposition obtained that is one order lower is said to be an instance of the proposition in quention. An instance is divided into zeroth-order, first-order, second-order, and third-order instance. A zeroth-order instance is also called a ground instance, because there is no variable in it. For example, zeroth-order fact proposition turn_around (Earth, Sun) can be regarded as a ground instance of first-order fact proposition turn_around(x, y) when x is assigned Earth, y Sun.

1.2.10 Mutually-inversistic sets revisited

A zeroth-order fact proposition can be regarded as a belonging-to relationship between a zeroth-level element constant and a zeroth-level set constant. For example, $1>0$ can be regarded as (1, 0) belonging to $x>y$, denoted by

$(1, 0)\in x>y$.

A first-order fact proposition is a zeroth-level set constant. For example, $x>y$ is a zeroth-order set constant constituted by all of the zeroth-level elements in Fig. 1.5 marked by △. The mutually inverse diagram of $x>y$ is composed of all of the vertexes in Fig. 1.5 marked by △.

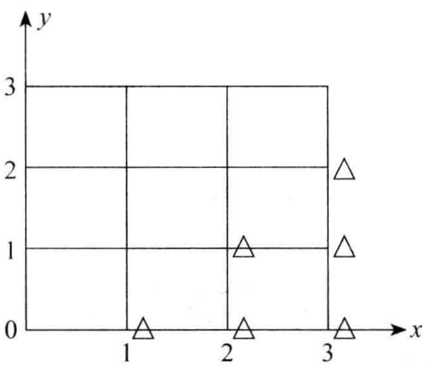


Fig. 1.5 The mutually inverse diagram of the zeroth-level set constant $x>y$

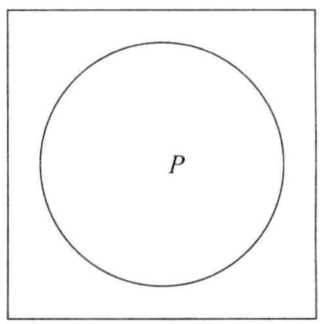


Fig. 1.6 The mutually inverse diagram of the zeroth-level set variable P

A second-order fact proposition is a zeroth-level set variable. For example, P is a zeroth-level set variable, its mutually inverse diagram is shown in Fig. 1.6.

A second-order single empirical or mathematical connection proposition is a first-level set constant. For example, $P\leqslant^{-1}Q$ is a first-level set constant, (man(x), mortal(x)) belongs to it, (positive_number(x), integer(x)) doesn't belong to it.

A first-order single empirical or mathematical connection proposition is a belonging-to relationship between a first-level element constant and a first-level set constant. For example, man(x) $\leqslant^{-1}$mortal(x) can be regarded as (man(x), mortal(x)) belonging to $P\leqslant^{-1}Q$, denoted by

(man(x), mortal(x))$\in P\leqslant^{-1}Q$.

A third-order single empirical or mathematical connection proposition is a first-level set variable. For example, Ψ is a first-level set variable.

A third-order single logical connection proposition is a second-level set constant. For example, $\Psi\leqslant^{-1}\Omega$ is a second-level set constant, $(P\leqslant^{-1}Q, \neg Q\leqslant^{-1}\neg P)$ belongs to it, $(P\leqslant^{-1}Q, P\leqslant^{-1}\neg Q)$ doesn't belong to it.

A second-order single logical connection proposition is a belonging-to relationship between a second-level element constant and a second-level set constant. For example, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ can be regarded as $(P\leqslant^{-1}Q, \neg Q\leqslant^{-1}\neg P)$ belonging to $\Psi\leqslant^{-1}\Omega$, denoted by

$(P\leqslant^{-1}Q, \neg Q\leqslant^{-1}\neg P)\in\Psi\leqslant^{-1}\Omega$.

A fourth-order single logical connection proposition is a second-level set variable. For example, Λ is a second-level set variable.

1.3 Simple-complex composition

A composition operator is used to make simple-complex composition. The composing propositions are called component propositions or simple propositions, the proposition composed is called a compound proposition or a complex proposition. For example, in composing parent(x, y)$\wedge$ancestor(y, z) from parent(x, y) and ancestor(y, z) by the fact composition operator $\wedge$, parent(x, y) and ancestor(y, z) are component propositions, parent(x, y)$\wedge$ancestor(y, z) is a compound proposition. The relationship between component propositions and compound propositions is relative. For example, $\{Q\vee R\}$ is a component proposition relative to $P\wedge\{Q\vee R\}$; it is a compound proposition relative to Q and R.

The principle of simple-complex composition (Frege's principle):

(i) The truth value of a compound proposition is determined by that of its n (n=1, 2) component proposition(s);

(ii) When the truth value(s) of its n component proposition(s) have been determined, that of the compound proposition is uniquely determined.

According to Frege's principle, we can give the operation tables of the various composition operators. Take negation $\neg$ as an example, its operation table is shown in Table 1.5.

Table 1.5 is so, because the binary tuple of a proposition and its truth value completely describe the proposition, and because negation $\neg$ operates on both the proposition and its

truth value. For example, we have (turn_around(Sun, Earth), F), and negation $\neg$ operates on both turn_around(Sun, Earth) and F, obtaining ($\neg$ turn_around(Sun, Earth), T).

Table 1.5
Operation table for $\neg$

(A, F)	($\neg$A, T)
(A, T)	($\neg$A, F)

Table 1.6
Truth table for $\neg$

tv(A)	tv($\neg$A)
F	T
T	F

Table 1.7
Wrong truth table

A	$\neg$A
F	T
T	F

Table 1.5 is redundant. The first components of the binary tuples of the argument are all A, and the first components of the binary tuples of the value are all $\neg$ A. To remove redundancy, we get the truth table Table 1.6. Table 1.6 cannot be simplified as Table 1.7. If so, then people will get a misconception that A is a propositional variable, F and T are propositional constants, A ranges over F and T.

Now, we give the truth tables of $\wedge$, $\vee$, and $\oplus$ in Tables 1.8 to 1.10.

Table 1.8
truth table for $\wedge$

tv(A)	tv(B)	tv(A$\wedge$B)
F	F	F
F	T	F
T	F	F
T	T	T

Table 1.9
truth table for $\vee$

tv(A)	tv(B)	tv(A$\vee$B)
F	F	F
F	T	T
T	F	T
T	T	T

Table 1.10
truth table for $\oplus$

tv(A)	tv(B)	tv(A$\oplus$B)
F	F	F
F	T	T
T	F	T
T	T	F

1.4 Zeroth-level predicate calculus

Zeroth-level predicate calculus studies the relationship between a term and a quasi-empirical or mathematical connection proposition.

If a proposition is a quasi-empirical or mathematical connection proposition, then at least one of the following holds:

(1) Namesake term variables occur on both sides of the binary predicates=, $<$, $\leqslant$, $>$, $\geqslant$; e.g., $x=x$, $x+y=y+x$. These term variables are called quasi-relevantly bound term variables.

(2) Namesake term variables occur on one side of the binary predicate=, $<$, $\leqslant$, $>$, $\geqslant$, but on both sides of the binary function, +, −, ∗, ÷; e.g., $x+(-x)=0$. These term variables are called quasi-intermediarily bound term variables.

(3) A free term variable and quasi-relevantly bound distinguished term 0 or 1 occur; e.g., $x*0=0$. (If a term variable occurs in a quasi-empirical or mathematical connection proposition only once, then it is free.)

A true quasi-empirical or mathematical connection proposition is called a quasi-empirical or mathematical theorem; a false one, a quasi-empirical or mathematical countertheorem.

One method to prove a quasi-empirical or mathematical theorem is mathematical induction, as shown in Example 1.1.

Example 1.1: Prove $1+3+5+\ldots+(2n-1)=n^2$ to be a quasi-empirical or mathematical theorem.

Proof: Basis: Let $n=1$, we obtain $1=1^2$.

Induction hypothesis: Suppose when $n=k$, we have $1+3+5+\ldots+(2k-1)=k^2$.

Induction step: When $n=k+1$, we have $1+3+5+\ldots+(2k-1)+(2k+1)=k^2+(2k+1)=(k+1)^2$.

The following are quasi-empirical or mathematical theorems in elementary algebra:

$x=x$,

$x+y=y+x$,

$x*y=y*x$,

$x+y+z=x+(y+z)$,

$x*y*z=x*(y*z)$,

$x*(y+z)=x*y+x*z$,

$x+0=x$,

$x*0=0$,

$x+1>x$,

$x*1=x$,

$x+(-x)=0$,

$x*x^{-1}=1$,

$-(-x)=x$.

Chapter 2
Human cognitive processes and basic principles of mutually-inversistic logic

2.1 Mutually inverse special propositions vs. mutually inverse general propositions

If part of a proposition is still a proposition, then it is called a subproposition of the proposition in question. For example, man(x) is a subproposition of man(x)$\leqslant^{-1}$mortal(x).

If the outmost operator of a proposition is a connection operator, then the proposition is a mutually inverse general proposition relative to the two subpropositions or their instances which the connection operator connects, and the two subpropositions or their instances which the connection operator connects are mutually inverse special propositions relative to the proposition in question. For example, man(x)$\leqslant^{-1}$mortal(x) is a mutually inverse general proposition relative to man(x) and mortal(x) or man(Aristotle) and mortal(Aristotle); and man(x) and mortal(x) or man(Aristotle) and mortal(Aristotle) are mutually inverse special propositions relative to man(x)$\leqslant^{-1}$mortal(x). For example, $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ is a mutually inverse general proposition relative to $P\leqslant^{-1}Q$ and $\neg Q\leqslant^{-1}\neg P$ or man(x)$\leqslant^{-1}$mortal(x) and $\neg$mortal(x)$\leqslant^{-1}\neg$man(x); and $P\leqslant^{-1}Q$ and $\neg Q\leqslant^{-1}\neg P$ or man(x)$\leqslant^{-1}$mortal(x) and $\neg$mortal(x)$\leqslant^{-1}\neg$man(x) are mutually inverse special propositions relative to $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$.

2.2 Unary cognitive processes

There are two unary cognitive processes for a proposition: establishment and employment. The establishment process is one that verifies, proves, disproves, or refutes a proposition. The employment process is one that uses an established proposition. For example, a mathematical theorem must be proved before it is used. Before a proposition is established, its truth value is not determined; at this time, it is called an unknown proposition. After a proposition is established, its truth value is determined, and at this time, it is called a known proposition. A known proposition is divided into a true proposition and a false one.

2.3 Binary cognitive processes

There exists certain relationship between establishment and employment. To establish an unknown proposition, we always employ other known propositions; and when we employ known propositions, we always do so to establish another unknown proposition. The relationship between establishment and employment is that human beings employ known propositions to establish unknown propositions. This relationship is one from the known to the unknown; it is binary cognitive process (see Fig. 2.1). Unary cognitive process refers to a proposition, while binary cognitive process refers to the relationship between propositions.

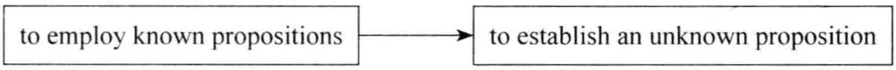


Fig. 2.1 Binary cognitive process

If from known propositions to unknown propositions is from mutually inverse special propositions to mutually inverse general propositions, then the corresponding binary cognitive process is called inductive composition (see Fig. 2.2).

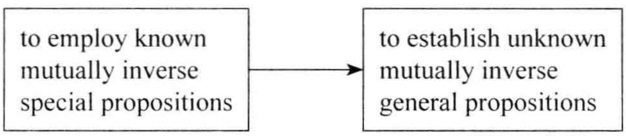


Fig. 2.2 Inductive composition

If from known propositions to unknown propositions is from mutually inverse general propositions to mutually inverse special propositions, then the corresponding binary cognitive process is called decomposition (see Fig. 2.3).

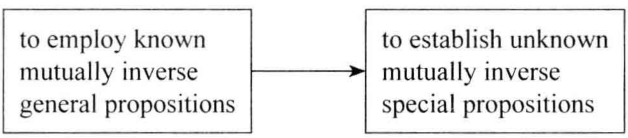


Fig. 2.3 Decomposition

Inductive composition and decomposition are mutually inverse binary cognitive processes, which justify the name mutually-inversistic logic.

The following are some examples of inductive composition and decomposition.

Example 2.1: When we use man(Aristotle) and mortal(Aristotle) to induce

man(Aristotle)$\leqslant^{-1}$mortal(Aristotle), use man(Russell) and mortal(Russell) to induce man(Russell)$\leqslant^{-1}$mortal(Russell); and use man(Aristotle)$\leqslant^{-1}$mortal(Aristotle) and man(Russell)$\leqslant^{-1}$mortal(Russell) to generalize man(x)$\leqslant^{-1}$mortal(x); what we do is first-level explicit inductive composition, because from man(Aristotle), mortal(Aristotle), man(Russell), mortal(Russell) to man(x)$\leqslant^{-1}$mortal(x) is from mutually inverse special proposition s to mutually inverse general proposition.

Example 2.2: Man(x) and mortal(x) are variable propositions; they do not have definite truth values. Let man(x) be true; we can infer that mortal(x) is bound to be true, thus we establish man(x)$\leqslant^{-1}$mortal(x). This is first-level implicit inductive composition, because from man(x) and mortal(x) to man(x)$\leqslant^{-1}$mortal(x) is from mutually inverse special propositions to mutually inverse general proposition.

Example 2.3: Suppose man(x)$\leqslant^{-1}$mortal(x) has been established, then from man(x)$\leqslant^{-1}$ mortal(x) and man(Zhou) to infer mortal(Zhou) is first-level decomposition, because man(x)$\leqslant^{-1}$mortal(x) is a mutually inverse general proposition relative to mortal(Zhou).

Example 2.4: When we use man(x)$\leqslant^{-1}$mortal(x) and $\neg$mortal(x)$\leqslant^{-1}\neg$man(x) to induce {man(x)$\leqslant^{-1}$mortal(x)}$\leqslant^{-1}${$\neg$mortal(x)$\leqslant^{-1}\neg$man(x)}, use integer(x)$\leqslant^{-1}$rational(x) and $\neg$rational(x)$\leqslant^{-1}\neg$integer(x) to induce {integer(x)$\leqslant^{-1}$ rational(x)}$\leqslant^{-1}${$\neg$rational(x) $\leqslant^{-1}\neg$integer(x)}; and use {man(x)$\leqslant^{-1}$mortal(x)}$\leqslant^{-1}${$\neg$mortal(x)$\leqslant^{-1}\neg$man(x)} and {integer(x)$\leqslant^{-1}$rational(x)}$\leqslant^{-1}${$\neg$rational(x)$\leqslant^{-1}\neg$integer(x)} to generalize $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$, what we do is second-level explicit inductive composition.

Example 2.5: $P\leqslant^{-1}Q$ and $\neg Q\leqslant^{-1}\neg P$ are variable propositions; they do not have definite truth values. Let $P\leqslant^{-1}Q$ be true; we can infer that $\neg Q\leqslant^{-1}\neg P$ is bound to be true, thus we establish $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$. This is second-level implicit inductive composition.

Example 2.6: Suppose $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ has been established, then from $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ and parent(x, y)$\leqslant^{-1}$ancestor(x, y) to infer $\neg$ancestor(x, y)$\leqslant^{-1}$ $\neg$parent(x, y) is second-level decomposition.

First-level inductive composition and decomposition are different scientific methods from induction and deduction, see Fig. 2.4.

first-level inductive composition	first-level explicit inductive composition	induction
	first-level implicit inductive composition	deduction
first-level decomposition	first-level decomposition	

Fig. 2.4 Different classifications of scientific methods

2.4 Man's cognitive route

Man's cognition of the universe starts from his own feeling of zeroth-order terms. Man feels the Sun, the Earth, Aristotle, Russell. Having felt zeroth-order terms, man discovers zeroth-order fact propositions. Man discovers that the Sun shines, then he establishes shine(Sun). Man discovers that Aristotle is a man, then he establishes man(Aristotle). Having established zeroth-order fact propositions, man makes first-level inductive compositions from zeroth-order fact propositions to empirical or mathematical theorems, as shown in Example 2.1; or from first-order fact propositions to empirical or mathematical theorems, as shown in Example 2.2. Example 2.1 is a preliminary stage of Example 2.2. Having established empirical or mathematical theorems, man's cognition diverges in two directions. One is first-level decomposition from empirical or mathematical theorems to zeroth-order fact propositions, as shown in Example 2.3. The other is second-level inductive composition from empirical or mathematical theorems to logical theorems, as shown in Example 2.4; or from second-order single empirical or mathematical connection propositions to logical theorems, as shown in Example 2.5. Example 2.4 is a preliminary stage of Example 2.5. Having established logical theorems, man makes second-level decompositions from it to empirical or mathematical theorems, as shown in Example 2.6. Unlike any other modern logics where a logical theorem is proved either by a logical axiomatic system or by a natural deduction system, in mutually-inversistic logic, a logical theorem is obtained by second-level inductive composition, as shown in Examples 2.4 and 2.5. Empirical or mathematical theorems can be obtained either from fact propositions by first-level inductive composition, as shown in Examples 2.1 and 2.2 or from logical theorems by second-level decomposition, as shown in Example 2.6. Zeroth-order fact propositions can be obtained in three ways: by being felt oneself, being informed by others, being inferred by first-level decomposition, as shown in Example 2.3. However, the beginning of the chain of being informed by others is still being felt oneself; therefore, for the whole mankind, zeroth-order fact propositions can be obtained in two ways: by being felt oneself or being inferred by first-level decomposition. For example, human beings feel themselves mortal(Aristotle); infer mortal(Zhou).

Western logics have logical axiomatic systems. A logical axiomatic system maintains a proof chain of logical theorems, the head of which is logical axioms. In mutually-inversistic logic, a single logical theorem is proved by second-level implicit inductive composition, starting from second-order single empirical or mathematical connection propositions, no need for logical axioms. The aim of proving a single logical theorem is to use it to infer a single empirical or mathematical theorem by second-level decomposition. Therefore, no proof chain of logical theorems is maintained.

Western mathematics has mathematical axiomatic systems. A mathematical axiomatic

system maintains a proof chain of mathematical theorems, the head of which is mathematical axioms. Although mutually-inversistic mathematics maintains a proof chain of mathematical theorems, the head of the chain is not mathematical axioms, but mathematical theorems proved by first-level implicit inductive composition. For example, in number systems, first, int(x)$\leqslant^{-1}$rat(x) and rat(x)$\leqslant^{-1}$real(x) are proved by first-level implicit inductive composition; then, they are taken as the heads of the proof chain and as the minor premise; and $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is the major premise; by second-level decomposition, int(x)$\leqslant^{-1}$ real(x) is inferred. Int(x) $\leqslant^{-1}$real(x) can then be used to infer new mathematical theorems.

Although mutually-inversistic logic and mathematics do not need axioms, for some disciplines it is not necessary to start the research from zeroth-order terms. So, in these disciplines it is convenient to introduce axioms as the starting point.

Occidental science is reduction theory, while oriental science is entirety theory. Mutual-inversism deems that empirical science, mathematics, and logic are a whole, they are inseparable. Occidental science is axiomatic, while oriental science is practical. Mutual-inversism is not axiomatic, it is developed in the need of automated theorem proving. Mutual-inversism is an oriental exact science essentially different from occidental science.

2.5 Classification of cognitive processes

From Figs. 2.2 and 2.3 we observe that a binary cognitive process links two unary cognitive processes; therefore, we can further classify the unary cognitive processes according to their positions in the binary cognitive process. The employment and establishment in Fig. 2.2 are linked by inductive composition; therefore, they are called employment by inductive composition and establishment by inductive composition, respectively. For example, the employment of man(Aristotle) and mortal(Aristotle) in Example 2.1 are employment by inductive composition; the establishment of man(x)$\leqslant^{-1}$mortal(x) in Example 2.1 is establishment by inductive composition. The employment and establishment in Fig. 2.3 are linked by decomposition; therefore, they are called employment by decomposition and establishment by decomposition, respectively. For example, the employment of man(x)$\leqslant^{-1}$mortal(x) and man(Zhou) in Example 2.3 are employment by decomposition; the establishment of mortal(Zhou) in Example 2.3 is establishment by decomposition.

Inductive composition is divided into explicit and implicit inductive composition. If an inductive composition depends on the assignments to the variables appearing in the propositions, then the inductive composition is explicit; otherwise, it is implicit. For example, the inductive composition of Example 2.1 is explicit because it depends on the assignments of Aristotle and Russell to the variable x; while the inductive composition of Example 2.2 is implicit because it doesn't depend on the assignments to x. Establishment is carried out in

two opposite ways: truthification (an English word coined by the author, meaning verification or proof) and falsification. Truthification is used to determine a proposition to be a true one, falsification a false one.

Establishment by decomposition is divided into truthification by decomposition (proof by decomposition) and falsification by decomposition. Employment by inductive composition is divided into employment by explicit inductive composition and employment by implicit inductive composition. Establishment by inductive composition is divided into truthification by inductive composition and falsification by inductive composition, which are further divided into truthification by explicit inductive composition, truthification by implicit inductive composition (proof by inductive composition), falsification by explicit inductive composition, and falsification by implicit inductive composition. The classification of unary cognitive processes is shown in Fig. 2.5. Fal.imp. ind.com. in Fig. 2.5 stands for falsification by implicit inductive composition, the rest can be inferred by analogy.

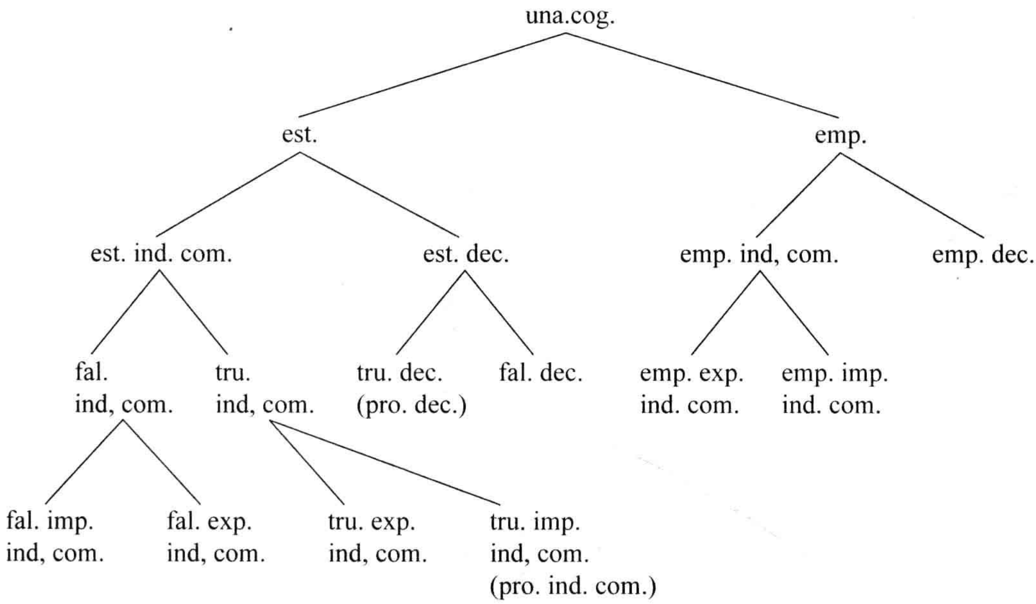


Fig. 2.5 Classification of unary cognitive processes

2.6 Inductive composition vs. decomposition

We take the explicit inductive composition of the unknown mutually inverse general proposition man(x)$\leqslant^{-1}$mortal(x) as an example to investigate inductive composition. Man(x) divides the term space into two sets: ⟨man(x), T⟩ and ⟨man(x), F⟩, mortal(x) into ⟨mortal(x), T⟩ and ⟨mortal(x), F⟩. Man(x) and mortal(x) jointly divide the term space into four minsets: (⟨man(x), F⟩⟨mortal(x), F⟩), (⟨man(x), F⟩⟨mortal(x), T⟩), (⟨man(x), T⟩⟨mortal(x), F⟩), (⟨man(x), T⟩⟨mortal(x), T⟩). See Fig. 2.6.

〈man(x), F〉 〈mortal(x), T〉	〈man(x), T〉 〈mortal(x), T〉	〈mortal(x), T〉
〈man(x), F〉 〈mortal(x), F〉	〈man(x), T〉 〈mortal(x), F〉	〈mortal(x), F〉
〈man(x), F〉	〈man(x), T〉	

Fig. 2.6 The four minsets of man(x)$\leqslant^{-1}$mortal(x)

Let us regard each minset as two arguments of a function, and stipulate a value for each of them, obtaining the operation table of an instance of mutually inverse implication, Table 2.1.

Table 2.1 Operation table of inductive composition of an instance of $\leqslant^{-1}$

〈man(x), F〉〈mortal(x), F〉	〈man(x)$\leqslant^{-1}$mortal(x), T〉
〈man(x), F〉〈mortal(x), T〉	〈man(x)$\leqslant^{-1}$mortal(x), n〉
〈man(x), T〉〈mortal(x), F〉	〈man(x)$\leqslant^{-1}$mortal(x), F〉
〈man(x), T〉〈mortal(x), T〉	〈man(x)$\leqslant^{-1}$mortal(x), T〉

Those minsets valued T constitute the support or truth area of man(x)$\leqslant^{-1}$mortal(x), because all of the assignments to x in these minsets are in support of it. The minset valued F constitutes the opposition or falsity area of man(x)$\leqslant^{-1}$mortal(x), because all of the assignments in this minset oppose it. The minset valued n constitutes the neutral or indeterminacy area of man(x)$\leqslant^{-1}$mortal(x), because the assignments in this minset neither support nor oppose it.

Table 2.1 is redundant; e.g., the first components of the binary tuple of the left argument are all man(x). To remove redundancy, we obtain the truth table Table 2.2.

Table 2.2 Truth table of inductive composition of an instance of $\leqslant^{-1}$

tv(man(x))	tv(mortal(x))	tv(man(x)$\leqslant^{-1}$mortal(x))
F	F	T
F	T	n
T	F	F
T	T	T

Now, let us assign the term constants in the term space to x, one by one. When x is assigned Aristotle, man(Aristotle) is true and mortal(Aristotle) is true. This assignment belongs to the minset (〈man(x), T〉〈mortal(x), T〉). According to the fourth row of Table 2.1, we obtain an induction operation:

〈man(Aristotle), T〉〈mortal(Aristotle), T〉	〈man(Aristotle)$\leqslant^{-1}$mortal(Aristotle), T〉

The second component of the value is T, which shows that "Aristotle"is an example supporting man(x)$\leqslant^{-1}$mortal(x) once. Likewise, when x is assigned Russell, we have an induction operation:

⟨man(Russell), T⟩ ⟨mortal(Russell), T⟩	⟨man(Russell)$\leqslant^{-1}$mortal(Russell), T⟩

Again, "Russell" is an example supporting man(x)$\leqslant^{-1}$mortal(x) once. The fourth row of Table 2.1 tells us three things: first, for each assignment of the term constant in the minset (⟨man(x), T⟩⟨mortal(x), T⟩) to x, the row represents an induction operation valued T; secondly, every such assignment is an example supporting man(x)$\leqslant^{-1}$mortal(x); thirdly, the minset (⟨man(x), T⟩⟨mortal(x), T⟩) is part of the support area of man(x)$\leqslant^{-1}$mortal(x).

When x is assigned a stone, man(stone) is false and mortal(stone) is false. This assignment belongs to the minset (⟨man(x), F⟩⟨mortal(x), F⟩). According to the first row of Table 2.1, we obtain an induction operation:

⟨man(stone), F⟩ ⟨mortal(stone), F⟩	⟨man(stone)$\leqslant^{-1}$mortal(stone), T⟩

The second component of the value is T, which shows that "stone" is an example supporting man(x)$\leqslant^{-1}$mortal(x) once. The same is true when x is assigned a brick. The first row of Table 2.1 tells us three things similar to those of the fourth row.

The third row of Table 2.1 tells us three things: first, when x was assigned a term constant in the minset (⟨man(x), T⟩⟨mortal(x), F⟩) (in this example, such an assignment doesn't exist), this row represents an induction operation valued F; secondly, every such assignment is a counterexample opposing man(x)$\leqslant^{-1}$mortal(x); thirdly, the minset (⟨man(x), T⟩⟨mortal(x), F⟩) constitutes the opposition area of man(x)$\leqslant^{-1}$mortal(x).

The second row of Table 2.1 tells us that the assignment of term constants such as a cat or a dog in the minset (⟨man(x), F⟩⟨mortal(x), T⟩) to x neither support nor oppose man(x)$\leqslant^{-1}$mortal(x). There is no need to determine whether there are such assignments or not; the minset constitutes the neutral area of man(x)$\leqslant^{-1}$mortal(x).

As x is assigned term constants in the term space, one by one, we obtain a series of induction operations, which constitute extended inductive inference of simple enumeration, so called because inductive inference of simple enumeration is from ⟨man(Aristotle)$\leqslant^{-1}$mortal(Aristotle), T⟩, ⟨man(Russell)$\leqslant^{-1}$mortal(Russell), T⟩, etc., to infer ⟨man(x)$\leqslant^{-1}$mortal(x), T⟩. However, here we must in addition consider ⟨man(stone)$\leqslant^{-1}$mortal(stone), T⟩, ⟨man(brick)$\leqslant^{-1}$mortal(brick), T⟩, etc. When x has been assigned all of the term constants in the term space, the inductive composition concludes. (If the term space is an infinite one, then with the finite life of man, he cannot make infinite assignments; instead he arbitrarily makes finite ones.) Now, in order to decide whether the unknown mutually inverse general proposition man(x)$\leqslant^{-1}$mortal(x) is truthified ("truthify" is the verb form of truthification) or falsified, we need to investigate whether the two criteria are met or not:

(1) All of the minsets that constitute the opposition area of the unknown mutually

inverse general proposition are empty;

(2) None of the minsets that constitute the support area of the unknown mutually inverse general proposition is empty.

If (1) is not met, i.e., there exists at least one instance in the opposition area, then the instance is a counterexample falsifying the unknown mutually inverse general proposition. If (2) is not met, i.e., at least one minset of the support area is empty, then we lack examples in the minset to support the unknown mutually inverse general proposition, which is also falsified. If both (1) and (2) are met, then the unknown mutually inverse general proposition is truthified. In this example, both criteria are met, therefore man(x)$\leqslant^{-1}$mortal(x) is truthified by explicit inductive composition.

Note that (1) and (2) are applicable not only to man(x)$\leqslant^{-1}$mortal(x), but also to any unknown mutually inverse general first-order single empirical or mathematical connection proposition and second-order single logical connection propositions; and not only to explicit inductive composition, but to implicit inductive composition as well.

After man(x)$\leqslant^{-1}$mortal(x) has been established, we can make decompositions using Table 2.3.(Please keep in mind that Table 2.3 can be simplified to truth table.)

Table 2.3 An instance of the operation table of decomposition for $\leqslant^{-1}$

⟨man(x) $\leqslant^{-1}$mortal(x), F⟩	⟨man(x), F⟩	⟨mortal(x), u⟩
⟨man(x) $\leqslant^{-1}$mortal(x), F⟩	⟨man(x), T⟩	⟨mortal(x), u⟩
⟨man(x) $\leqslant^{-1}$mortal(x), T⟩	⟨man(x), F⟩	⟨mortal(x), u⟩
⟨man(x) $\leqslant^{-1}$mortal(x), T⟩	⟨man(x), T⟩	⟨mortal(x), T⟩

⟨man(x)$\leqslant^{-1}$mortal(x), T⟩ serves as the major premise; ⟨man(Zhou), T⟩ is the minor premise. From the major premise and minor premise we can decompose the conclusion ⟨mortal(Zhou), T⟩. However, when man(x)$\leqslant^{-1}$mortal(x) is true and man(x) is false, then mortal(x) cannot be uniquely determined. Because x can be a cat, in this case, mortal(cat) is true. X can also be a stone; in this case, mortal(stone) is false. Therefore, decomposition just cannot go on.

2.7 The Principle of inductive composition, the principle of decomposition, and the principle of mutual inverseness between inductive composition and decomposition

If the outmost connection operator of a proposition is a multi-cellular connection operator, then the proposition is called a multi-cellular proposition. If the outmost

connection operator of a proposition is a unicellular connection operator, then the proposition is called a unicellular proposition.

The principle of inductive composition:

(1) The truth value of an unknown mutually inverse general proposition is determined by that of its two known mutually inverse special propositions;

(2) When determining the truth value of an unknown unicellular mutually inverse general proposition, we must investigate every minset of it;

(3) When determining the truth value of an unknown multi-cellular mutually inverse general proposition, we need not investigate some minsets of it.

The principle of decomposition:

(1) The truth value of an unknown mutually inverse special proposition is determined by that of a known mutually inverse general proposition and its known mutually inverse special proposition;

(2) When the truth values of the known mutually inverse general proposition and its known mutually inverse special proposition have been determined, in some cases the truth value of the unknown mutually inverse special proposition cannot be uniquely determined. These cases are the invalid inference forms, and in these cases, inference cannot go on;

(3) When the truth values of the known mutually inverse general proposition and its known mutually inverse special proposition have been determined, in some cases the truth value of the unknown mutually inverse special proposition can be uniquely determined. These cases are the valid inference forms, and in these cases, inference can go on.

The principle of mutual inverseness between inductive composition and decomposition:

(1) When an inductive composition concludes, the unknown mutually inverse general proposition becomes the known one, then decomposition can begin;

(2) When a decomposition concludes, the unknown mutually inverse special proposition becomes the known one, then new inductive composition can begin;

(3) Inductive composition and decomposition are mutually inverse human binary cognitive processes.

2.8 Truth tables of inductive composition and decomposition for the connection operators

First, let us discuss why the truth table of inductive composition for $\leqslant^{-1}$ is defined as Table 1.2. A$\leqslant^{-1}$B means that A is a sufficient condition of B. A$<^{-1}$B means that A is a sufficient but not necessary condition of B. A$=^{-1}$B means that A is a sufficient and necessary

condition of B. $A \leqslant^{-1} B$ is equivalent to the exclusive or of $A <^{-1} B$ and $A =^{-1} B$. The truth tables for $<^{-1}$ are shown in Tables. 2.4 to 2.6.The trath tables for $=^{-1}$ are shown in Tables 2.7 to 2.9.

Table 2.4 is the same as the truth table for $\rightarrow$. The mutually inverse diagram of $<^{-1}$ is the same as Fig. 1.1 (d). From Fig. 1.1 (d) we see that when A is true, B is bound to be true; while when B is true, A may not be true. This is to say, $A <^{-1} B$ indeed represents that A is a sufficient but not necessary condition of B.

Table 2.4
Inductive composition for $<^{-1}$

tv(A)	tv(B)	tv($A <^{-1} B$)
F	F	T
F	T	T
T	F	F
T	T	T

Table 2.5
Decomposition one for $<^{-1}$

tv($A <^{-1} B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	u
T	T	T

Table 2.6
Decomposition two for $<^{-1}$

tv($A <^{-1} B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	F
T	T	u

The truth tables for $=^{-1}$ are shown in Tables 2.7 to 2.9.

Table 2.7
Inductive composition for $=^{-1}$

tv(A)	tv(B)	tv($A =^{-1} B$)
F	F	T
F	T	F
T	F	F
T	T	T

Table 2.8
Decomposition one for $=^{-1}$

tv($A =^{-1} B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	F
T	T	T

Table 2.9
Decomposition two for $=^{-1}$

tv($A =^{-1} B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	F
T	T	T

Table 2.7 is the same as the truth table for $\leftrightarrow$. The mutually inverse diagram of $=^{-1}$ is the same as Fig. 1.1 (e). Obviously, $A =^{-1} B$ represents that A is a sufficient and necessary condition of B.

The common part of $A <^{-1} B$ and $A =^{-1} B$ is that in both cases A is a sufficient condition of B. The common part of Tables 2.4 and 2.7 is that in both tables the first row is T, the third row is F, the fourth row is T. $A \leqslant^{-1} B$ is the common part of $A <^{-1} B$ and $A =^{-1} B$: A is a sufficient condition of B. Hence, in Table 1.2, the first row is T, the third row is F, the fourth row is T. The different part of $A <^{-1} B$ and $A =^{-1} B$ is that in the $A <^{-1} B$ case, B is not a sufficient condition of A; while in the $A =^{-1} B$ case, B is a sufficient condition of A. The different part of Tables 2.4 and 2.7 lies in their second row. Since $A \leqslant^{-1} B$ does not care whether B is a sufficient condition of A or not, the second row of Table 1.2 is n, meaning "need not determine whether it is true or false".

The connection operators $\leqslant^{-1}$, $=^{-1}$, and $<^{-1}$ in a mutually inverse general proposition connects two mutually inverse special propositions, the left one is the antecedent, the right

one is the consequent.

The truth tables of decomposition one for $\leqslant^{-1}$, $=^{-1}$, and $<^{-1}$ are the affirmative expressions of hypothetical inference. The truth tables of decomposition two for $\leqslant^{-1}$, $=^{-1}$, and $<^{-1}$ are the negative expressions of hypothetical inference.

The establishment of A being a sufficient condition of B is that suppose A is true, if we can infer that B is true, then we establish that A is a sufficient condition of B. The employment of A being a sufficient condition of B is that suppose A is a sufficient condition of B, then from A being true we can infer B being true.

$A\vee/^{-1}B$ is equivalent to $\neg A\leqslant^{-1}B$, $A\vee^{-1}B$ to $\neg A<^{-1}B$, $A\oplus^{-1}B$ to $\neg A=^{-1}B$. And $A\vee/^{-1}B$ is the exclusive or of $A\vee^{-1}B$ and $A\oplus^{-1}B$. The truth table of inductive composition for $A\vee/^{-1}B$ can be obtained from that for $\neg A\leqslant^{-1}B$ in this way: rewrite Table 1.2 as Table 2.10, which is the same as Table 2.11; interchange the first row and the third row, interchange the second row and the fourth row of Table 2.11, obtaining Table 2.12, which is the same as Table 2.13. The truth tables of decomposition for $A\vee/^{-1}B$, shown in Tables 2.14 and 2.15, can be obtained similarly.

Table 2.10
From $\neg A\leqslant^{-1}B$ to $A\vee/^{-1}B$

tv($\neg$A)	tv(B)	tv($\neg A\leqslant^{-1}B$)
F	F	T
F	T	n
T	F	F
T	T	T

Table 2.11
From $\neg A\leqslant^{-1}B$ to $A\vee/^{-1}B$

tv(A)	tv(B)	tv($\neg A\leqslant^{-1}B$)
T	F	T
T	T	n
F	F	F
F	T	T

Table 2.12
From $\neg A\leqslant^{-1}B$ to $A\vee/^{-1}B$

tv(A)	tv(B)	tv($\neg A\leqslant^{-1}B$)
F	F	F
F	T	T
T	F	T
T	T	n

Table 2.13
Inductive composition for $\vee/^{-1}$

tv(A)	tv(B)	tv($A\vee/^{-1}B$)
F	F	F
F	T	T
T	F	T
T	T	n

Table 2.14
Decomposition one for $\vee/^{-1}$

tv($A\vee/^{-1}B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	T
T	T	u

Table 2.15
Decomposition two for $\vee/^{-1}$

tv($A\vee/^{-1}B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	T
T	T	u

Now, we give the truth tables for $\vee^{-1}$ in Tables 2.16 to 2.18, those for $\oplus^{-1}$ in Tables 2.19 to 2.21.

Table 2.16

Inductive composition for $\vee^{-1}$

tv(A)	tv(B)	tv($A\vee^{-1}B$)
F	F	F
F	T	T
T	F	T
T	T	T

Table 2.17

Decomposition one for $\vee^{-1}$

tv($A\vee^{-1}B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	T
T	T	u

Table 2.18

Decomposition two for $\vee^{-1}$

tv($A\vee^{-1}B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	T
T	T	u

Table 2.19

Inductive composition for $\oplus^{-1}$

tv(A)	tv(B)	tv($A\oplus^{-1}B$)
F	F	F
F	T	T
T	F	T
T	T	F

Table 2.20

Decomposition one for $\oplus^{-1}$

tv($A\oplus^{-1}B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	T
T	T	F

Table 2.21

Decomposition two for $\oplus^{-1}$

tv($A\oplus^{-1}B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	T
T	T	F

The truth tables of decomposition for $\vee/^{-1}$, $\vee^{-1}$, and $\oplus^{-1}$ can be used to make disjunctive inference.

The truth tables for $/\wedge^{-1}$ are given in Tables 2.22 to 2.24, those for $\times^{-1}$ are given in Tables 2.25 to 2.27.

Table 2.22

Inductive composition for $/\wedge^{-1}$

tv(A)	tv(B)	tv($A/\wedge^{-1}B$)
F	F	n
F	T	n
T	F	n
T	T	T

Table 2.23

Decomposition one for $/\wedge^{-1}$

tv($A/\wedge^{-1}B$)	tv(A)	tv(B)
F	F	u
F	T	F
T	F	u
T	T	u

Table 2.24

Decomposition two for $/\wedge^{-1}$

tv($A/\wedge^{-1}B$)	tv(B)	tv(A)
F	F	u
F	T	F
T	F	u
T	T	u

Table 2.25

Inductive composition for $\times^{-1}$

tv(A)	tv(B)	tv($A\times^{-1}B$)
F	F	T
F	T	T
T	F	T
T	T	T

Table 2.26

Decomposition one for $\times^{-1}$

tv($A\times^{-1}B$)	tv(A)	tv(B)
F	F	u
F	T	u
T	F	u
T	T	u

Table 2.27

Decomposition two for $\times^{-1}$

tv($A\times^{-1}B$)	tv(B)	tv(A)
F	F	u
F	T	u
T	F	u
T	T	u

At a first glance, it seems that these operations are not closed, because the arguments participating in the operations are F and T; while the values yielded are F, T, n, and u. But

actually, these operations are closed, because n and u will not be generated. We take Table 2.1 as an example for n. Suppose x is assigned a cat, after finding the result is n, the assignment cat is discarded, $\langle man(cat) \leqslant^{-1} mortal(cat), n\rangle$ is not generated. We take Table 2.3 as an example of u. Suppose $man(x) \leqslant^{-1} mortal(x)$ is true and $man(x)$ is false, after finding the result is u, no inference is made, $\langle mortal(x), u\rangle$ is not generated.

Note that the truth tables of decomposition of these connection operators are the generalized inverse functions of their truth tables of inductive composition, because for multi-cellular connection operators, their inverse functions in strict sense do not exist, for unicellular connection operators, their inverse functions in strict sense exist but are unworkable.

2.9 Mutually inverse diagrams for the connection operators

We can use mutually inverse diagrams to denote the relationship between two sets. Suppose A and B are two sets that are not distinguished sets, then we can construct mutually inverse diagrams for the 7 unicellular propositions shown in Figs. 2.7 to 2.13.

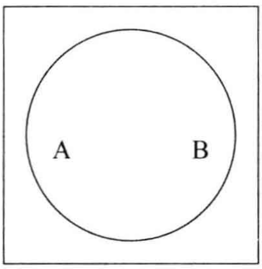


Fig. 2.7
$A =^{-1} B$ and $\neg A \oplus^{-1} B$

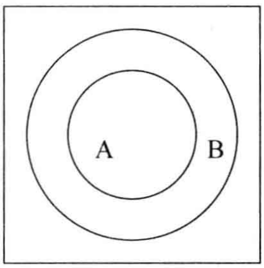


Fig. 2.8
$A <^{-1} B$ and $\neg A \vee^{-1} B$

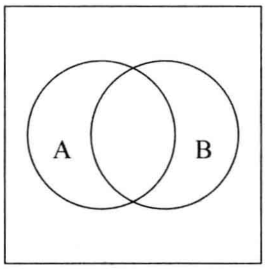


Fig. 2.9
$A \times^{-1} B$

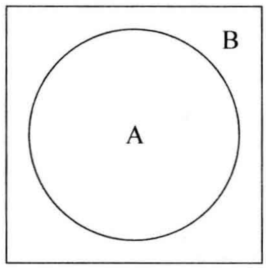


Fig. 2.10
$A \oplus^{-1} B$ and $\neg A =^{-1} B$

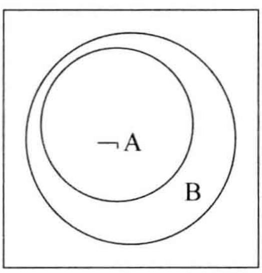


Fig. 2.11
$\neg A <^{-1} B$ and $A \vee^{-1} B$

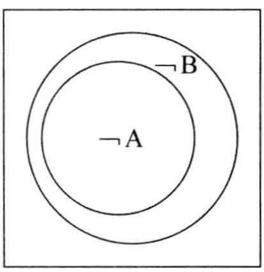


Fig. 2.12
$\neg A <^{-1} \neg B$ and $A \vee^{-1} \neg B$

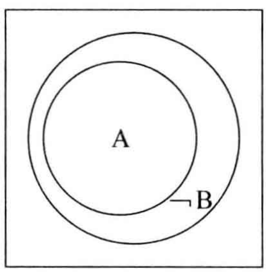


Fig. 2.13
$A <^{-1} \neg B$ and $\neg A \vee^{-1} \neg B$

The unicellular proposition $A=^{-1}B$ being true is denoted by Fig. 2.7. $A=^{-1}B$ being false is denoted by Figs. 2.8 to 2.13. The rest can be inferred by analogy. The multi-cellular proposition $A\leqslant^{-1}B$ is denoted by Figs. 2.7 and 2.8, the common characteristic of which is the absence of the minset (<A, T><B, F>). The multi-cellular proposition $A/\wedge^{-1}B$ is denoted by Figs. 2.7 to 2.9 and 2.11 to 2.12, the common characteristic of which is the presence of the minset (<A, T><B, T>). The multi-cellular proposition $A\vee/^{-1}B$ is denoted by Figs. 2.10 and 2.11, the common characteristic of which is the absence of the minset (<A, F><B, F>).

The relationship between $A\wedge B$ and $A/\wedge^{-1}B$ is that if $A\wedge B$ is true, i.e., the minset (<A, T><B, T>) is nonempty, then $A/\wedge^{-1}B$ is true; otherwise, $A/\wedge^{-1}B$ is false. $\neg\{A\wedge B\}$ is equivalent to $\neg A\vee\neg B$. $\neg\{A/\wedge^{-1}B\}$ is equivalent to $\neg A\vee/^{-1}\neg B$. The relationship between $\neg A\vee\neg B$ and $\neg A\vee/^{-1}\neg B$ is that if $\neg A\vee\neg B$ is true, then $\neg A\vee/^{-1}\neg B$ is true; otherwise, $\neg A\vee/^{-1}\neg B$ is false.

2.10 The principle of meaningfulness and meaninglessness duality for the distinguished propositions

$\varnothing_0$, U_0, $\varnothing_1$, U_1, $\varnothing_2$, U_2, are distinguished propositions or distinguished sets. The empty sets do not reflect objective being; e.g., dragon(x) is an empty set, for there is no dragon at all in the universe. The universal sets reflect ubiquity. However, in the abstract logical operations, distinguished propositions are bound to be yielded, and are bound to be operated upon; e.g., the conjunction of P and $\neg$P is $\varnothing_0$.

Single empirical or mathematical connection propositions and quasi-logical connection propositions with the outmost connection operator being $\leqslant^{-1}$ study the transition from one objectively existing but not ubiquitous fact to another. The two facts that the empirical or mathematical connection operator connects must not be distinguished propositions. The multiple empirical or mathematical connection propositions study the connections among the objectively existing but not ubiquitous facts which must not be distinguished propositions. Single logical connection propositions and quasi-transcendent logical connection propositions with the outmost connection operator being $\leqslant^{-1}$ study the transition from one objectively existing but not ubiquitous single empirical or mathematical connection to another. The two single empirical or mathematical connection propositions that the logical connection operator connects must not be distinguished propositions.

The quasi-logical connection proposition with the outmost connection operator being $=^{-1}$ and the quasi-transcendent logical connection proposition with the outmost connection operator being $=^{-1}$ represent abstract operations. There can be distinguished propositions in them. In fact, their uses are as replacements. For example, $P\wedge Q$ in the proposition $P\wedge Q$

$\leqslant^{-1}R$ can be replaced by $Q\wedge P$ in the quasi-logical connection proposition $P\wedge Q=^{-1} Q\wedge P$, thus $P\wedge Q\leqslant^{-1}R$ is equivalently transformed into $Q\wedge P\leqslant^{-1}R$.

The principle of meaningfulness and meaninglessness duality for the distinguished propositions:

(1) The distinguished proposition occurring in single empirical or mathematical connection propositions, multiple empirical or mathematical connection propositions, single logical connection propositions, quasi-logical connection propositions with the outmost connection operator being $\leqslant^{-1}$, and quasi-transcendent logical connection propositions with the outmost connection operator being $\leqslant^{-1}$ is meaningless; one occurring in quasi-logical connection propositions with the outmost connection operator being $=^{-1}$ and quasi-transcendent logical connection proposition with the outmost connection operator being $=^{-1}$ is meaningful;

(2) If there is a distinguished proposition occurring in single empirical or mathematical connection propositions, multiple empirical or mathematical connection propositions, single logical connection propositions, quasi-logical connection propositions with the outmost connection operator being $\leqslant^{-1}$, and quasi-transcendent logical connection propositions with the outmost connection operator being $\leqslant^{-1}$, then these propositions are meaningless propositions; otherwise, they are meaningful ones;

(3) Quasi-logical connection propositions with the outmost connection operator being $=^{-1}$ and quasi-transcendent logical connection proposition with the outmost connection operator being $=^{-1}$ are meaningful propositions regardless of whether or not there is a distinguished proposition in them.

Chapter 3
First-level single quasi-predicate calculus

First-level single quasi-predicate calculus studies the relationship between fact propositions and single empirical or mathematical connection propositions, quasi-logical connection propositions.

3.1 Meaningless and meaningful first-order single empirical or mathematical connection propositions

$X=y \wedge x<y$ in the proposition $x=y \wedge x<y \leq^{-1} x \leq y$ is an empty set, according to the principle of meaningfulness and meaninglessness duality for distinguished propositions, the proposition is a meaningless one; while parent(x, y)$\leq^{-1}$ancestor(x, y) is a meaningful proposition, for none of parent(x, y) and ancestor(x, y) is a distinguished proposition. For an implication proposition to be a meaningful one, it is sufficient to require that the antecedent be not permanently false and the consequent be not permanently true.

3.2 Free and bound first-order single empirical or mathematical connection propositions

Unlike other first-order logics, the first-level single quasi-predicate calculus is quantifier-free, the boundness of term variables is realized by the common occurrences of term variables in a single empirical or mathematical connection proposition.

A single empirical or mathematical connection proposition is a connection between two fact propositions, and is connected by the same terms. For example, the fact propositions man(Aristotle) and mortal(Aristotle) in man(Aristotle)$\leq^{-1}$mortal(Aristotle) are connected by the same term Aristotle, while there is no connection between man(Aristotle) and mortal(Russell) in man(Aristotle)$\leq^{-1}$mortal(Russell). Generally, the fact propositions man(x) and mortal(x) in man(x) $\leq^{-1}$mortal(x) are connected by the same term variable x, while there is no connection between man(x) and mortal(y) in man(x)$\leq^{-1}$mortal(y), because there is not the same term variable to connect them. Namesake term variables occurring on both sides of an empirical or mathematical connection operator are called relevantly bound term variables; e.g., x in man(x)$\leq^{-1}$mortal(x) is a relevantly bound term variable.

Sometimes, although there are the same terms between two fact propositions, there

is still no connection between them. For example, there is no connection between the fact propositions parent(Old John, Big John)$\wedge$ancestor(Middle John, Little John) and ancestor(Old John, Little John) in parent(Old John, Big John)$\wedge$ancestor(Middle John, Little John)$\leqslant^{-1}$ancestor(Old John, Little John), although there are the same terms, Old John and Little John, between them. Generally, there is no connection between the fact propositions parent(x, y)$\wedge$ancestor(w, z) and ancestor(x, z) in parent(x, y)$\wedge$ancestor(w, z)$\leqslant^{-1}$ancestor(x, z), although there are the same term variables, x and z, between them. While there is a connection between the fact propositions parent(Old John, Big John)$\wedge$ancestor(Big John, Little John) and ancestor(Old John, Little John) in parent(Old John, Big John)$\wedge$ancestor(Big John, Little John)$\leqslant^{-1}$ancestor(Old John, Little John), this is not only because there are the same terms, Old John and Little John, to establish a connection between them but also because there is Big John to act as an intermediary role to the connection established by Old John and Little John. Generally, there is a connection between the fact propositions parent(x, y)$\wedge$ancestor(y, z) and ancestor(x, z) in parent(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z), this is not only because there are the same term variables, x and z, to establish a connection between them but also because there is y to act as an intermediary role to the connection established by x and z. Y in parent(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z) is called an intermediarily bound term variable. A term variable is intermediarily bound if it occurs on one side of an empirical or mathematical connection operator but on both sides of a binary fact composition operator.

If a term variable occurs on one side of an empirical or mathematical connection operator but on both sides of a binary predicate, then it is an additionally bound term variable. For example, z in $x+z=y+z\leqslant^{-1}x=y$ is an additionally bound term variable.

If a term variable occurs on one side of a binary predicate but on both sides of a binary function, then it is a juxtaposed bound term variable. For example, z in $x*x+y*y+z*z\leqslant 1\leqslant^{-1}$ $x*x+y*y\leqslant 1$ is a juxtaposed bound term variable.

If a term variable occurs in a single empirical or mathematical connection proposition only once, then it is a free term variable. For example, x and y in man(x) $\leqslant^{-1}$mortal(y) are all free term variables. If a term variable occurs in a single empirical or mathematical connection proposition more than once, then it is bound, for it can be at least one of the four above mentioned boundness. Take $x<y\wedge x<z\leqslant^{-1}x<y+z$ as an example. The first and second occurrences of x are intermediarily bound, the first and third occurrences of x are relevantly bound, the second and third occurrences of x are relevantly bound.

If all of the term variables in a single empirical or mathematical connection proposition are bound, then the proposition is bound. If at least one term variable in a single empirical or mathematical connection proposition is free, then the proposition is free.

If a first-order single empirical or mathematical connection proposition is meaningful and

bound, then the establishment by inductive composition of it can be made. For an implication proposition, meaningfulness and boundness means that it is not an implication paradox, for meaningfulness means that the antecedent is not permanently false, the consequent is not permanently true; boundness means that the antecedent and the consequent share the same relevantly bound term variable.

3.3 First-level explicit inductive composition

Example 3.1: Truthify man(x)$\leqslant^{-1}$mortal(x) by explicit inductive composition.

Solution: Man(x) and mortal(x) are neither empty sets nor universal sets. Therefore, man(x)$\leqslant^{-1}$mortal(x) is a meaningful proposition. X in man(x)$\leqslant^{-1}$mortal(x) is relevantly bound. Therefore, man(x)$\leqslant^{-1}$mortal(x) is a bound proposition. Inductive composition can go on. From Table 1.2 we learn that the minset (⟨man(x), T⟩⟨mortal(x), F⟩) belongs to the opposition area of man(x)$\leqslant^{-1}$mortal(x), the minsets (⟨man(x), F⟩⟨mortal(x), F⟩) and (⟨man(x), T⟩⟨mortal(x), T⟩) belong to the support area of man(x)$\leqslant^{-1}$mortal(x). The two criteria for deciding whether man(x)$\leqslant^{-1}$mortal(x) is truthified or falsified are given below:

(1) (⟨man(x), T⟩⟨mortal(x), F⟩) is empty;

(2) None of (⟨man(x), F⟩⟨mortal(x), F⟩) and (⟨man(x), T⟩⟨mortal(x), T⟩) is empty.

Criterion (1) is met; i.e., (⟨man(x), T⟩⟨mortal(x), F⟩) is empty, for there is no immortal man on the Earth. We would like to point out that if (1) is met, then (2) is automatically met; i.e., if (⟨man(x), T⟩⟨mortal(x), F⟩) is empty, then none of (⟨man(x), F⟩⟨mortal(x), F⟩) and (⟨man(x), T⟩⟨mortal(x), T⟩) is empty. Because if (⟨man(x), T⟩⟨mortal(x), T⟩) is also empty, then the term space is shown in Fig. 3.1, where the shaded minsets represent empty ones.

⟨man(x), F⟩ ⟨mortal(x), T⟩	⟨man(x), T⟩ ⟨mortal(x), T⟩	⟨mortal(x), T⟩
⟨man(x), F⟩ ⟨mortal(x), F⟩	⟨man(x), T⟩ ⟨mortal(x), F⟩	⟨mortal(x), F⟩
⟨man(x), F⟩	⟨man(x), T⟩	

Fig. 3.1 Both (⟨man(x), T⟩⟨mortal(x), F⟩) and (⟨man(x), T⟩⟨mortal(x), T⟩) are empty

From Fig. 3.1 we observe that man(x) is an empty set or permanently false proposition. According to the principle of meaningfulness and meaninglessness duality for distinguished propositions, man(x)$\leqslant^{-1}$mortal(x) is a meaningless proposition, which contradicts the fact that man(x)$\leqslant^{-1}$mortal(x) is a meaningful one. Therefore, (⟨man(x), T⟩⟨mortal(x), T⟩) cannot be empty. If (⟨man(x), F⟩⟨mortal(x), F⟩) is also empty, then the term space is shown in Fig. 3.2, where the shaded minsets represent empty ones.

⟨man(x), F⟩ ⟨mortal(x), T⟩	⟨man(x), T⟩ ⟨mortal(x), T⟩	⟨mortal(x), T⟩
⟨man(x), F⟩ **⟨mortal(x), F⟩**	**⟨man(x), T⟩** **⟨mortal(x), F⟩**	⟨mortal(x), F⟩
⟨man(x), F⟩	⟨man(x), T⟩	

Fig. 3.2 Both (⟨man(x), T⟩⟨mortal(x), F⟩) and (⟨man(x), F⟩⟨mortal(x), F⟩) are empty

From Fig.3.2 we observe that mortal(x) is a universal set or permanently true proposition. According to the principle of meaningfulness and meaninglessness duality for distinguished propositions, man(x)$\leqslant^{-1}$mortal(x) is a meaningless proposition, which contradicts the fact that man(x)$\leqslant^{-1}$mortal(x) is a meaningful one. Therefore, (⟨man(x), F⟩⟨mortal(x), F⟩) cannot be empty.

Thus, we conclude that when establishing man(x)$\leqslant^{-1}$mortal(x) by inductive composition, we need only to investigate the minset (⟨man(x), T⟩⟨mortal(x), F⟩), and the two criteria become one; that is criterion (1): if there is at least one assignment to x belonging to (⟨man(x), T⟩⟨mortal(x), F⟩), then man(x)$\leqslant^{-1}$mortal(x) is falsified; otherwise, it is truthified.

Example 3.2: Truthify man(x)$/\wedge^{-1}$land_on_the_Moon(x) by explicit inductive composition.

Solution: Man(x) and land_on_the_Moon(x) are neither empty sets nor universal sets, therefore, man(x)$/\wedge^{-1}$land_on_the_Moon(x) is meaningful. X in man(x)$/\wedge^{-1}$land_on_the_Moon(x) is relevantly bound, therefore, the proposition is bound. Inductive composition can go on. From Table 2.22, we learn that the minset (⟨man(x), T⟩⟨land_on_the_Moon(x), T⟩) constitutes the support area of man(x)$/\wedge^{-1}$land_on_the_Moon(x), no minset constitutes the opposition area. Therefore, we need only to investigate criterion (2). Astronaut N. Armstrong belongs to the minset (⟨man(x), T⟩⟨land_on_the_Moon(x), T⟩), which is not empty. Thus, man(x)$/\wedge^{-1}$land_on_the_Moon(x) is truthified by explicit inductive composition.

Example 3.3: Truthify publish(x)$\vee/^{-1}$perish(x) by explicit inductive composition.

Solution: Publish(x) and perish(x) are neither empty sets nor universal sets, therefore, publish(x)$\vee/^{-1}$perish(x) is meaningful. X in publish(x)$\vee/^{-1}$perish(x) is relevantly bound, therefore, the proposition is bound. Inductive composition can go on. From Table 2.13 we learn that the minset (⟨publish(x), F⟩⟨perish(x), F⟩) constitutes the opposition area of publish(x)$\vee/^{-1}$perish(x), and that the minsets (⟨publish(x), F⟩⟨perish(x), T⟩)(⟨publish(x), T⟩⟨perish(x), F⟩) constitute the support area of publish(x)$\vee/^{-1}$perish(x). There is no such professor who does not publish and who does not perish; i.e., the minset (⟨publish(x), F⟩⟨perish(x), F⟩) is empty. Because of this, we assert, similar to the analysis given in Example 3.1, that none of (⟨publish(x), F⟩⟨perish(x), T⟩)(⟨publish(x), T⟩⟨perish(x), F⟩) is empty.

Therefore, publish(x)$\vee/^{-1}$perish(x) is truthified by explicit inductive composition.

Example 3.4: Falsify prime_number(x)$\leqslant^{-1}$odd_number(x) by explicit inductive composition.

Solution: Prime_number(x) and odd_number(x) are neither empty sets nor universal sets, therefore, prime_number(x)$\leqslant^{-1}$odd_number(x) is meaningful. X in prime_number(x)$\leqslant^{-1}$odd_number(x) is relevantly bound, therefore, the proposition is bound. Inductive composition can go on. 2 belongs to the minset (<prime_number(x), T><odd_number(x), F>), which constitutes the opposition area of prime_number(x)$\leqslant^{-1}$odd_number(x). Thus, 2 serves as a counterexample, falsifying prime_number(x)$\leqslant^{-1}$odd_number(x).

3.4 First-level implicit inductive composition

Example 3.5: Truthify man(x)$\leqslant^{-1}$mortal(x) by implicit inductive composition.

Solution: Man(x)$\leqslant^{-1}$mortal(x) is meaningful and bound, inductive composition can go on. Suppose x is a man, then x is bound to experience birth, infancy, growing-up, adult, senility, death. That is, the minset (<man(x), T><mortal(x), F>) that constitutes the opposition area of man(x)$\leqslant^{-1}$mortal(x) is empty, and criterion (1) is met. Similar to the analysis given in Example 3.1, none of the minsets (<man(x), F><mortal(x), F>) and (<man(x), T><mortal(x), T>) is empty, and criterion (2) is also met. Thus, man(x)$\leqslant^{-1}$mortal(x) is truthified by implicit inductive composition.

Example 3.6: Truthify man(x)$/\wedge^{-1}$live_extraterrestrially(x) by implicit inductive composition.

Solution: Man(x) and live_extraterrestrially(x) are neither empty sets nor universal sets, therefore, man(x)$/\wedge^{-1}$live_extraterrestrially(x) is meaningful. X in man(x)$/\wedge^{-1}$live_extraterrestrially(x) is relevantly bound, therefore, the proposition is bound. Inductive composition can go on. The minset (<man(x), T>< live_extraterrestrially(x), T>) constitutes the support area of man(x)$/\wedge^{-1}$live_extraterrestrially(x). At present, we have not found an extraterrestrial; i.e., we have not found a term constant belonging to this minset. However, in every 100,000 planets there is one whose condition is similar to that of the Earth. Therefore, the possibility of such a term constant exists. Thus, man(x)$/\wedge^{-1}$live_extraterrestrially(x) is truthified by implicit inductive composition.

The following are single empirical or mathematical theorems in elementary algebra:

$x=y=^{-1}y=x$

$x<y=^{-1}y\geqslant x$

$x=y\wedge y=z\leqslant^{-1}x=z$

$x<y\wedge y<z\leqslant^{-1}x<z$

$x=y\wedge y<z\leqslant^{-1}x<z$

$x<y\leqslant^{-1}x\leqslant y$

$x\leqslant y\leqslant^{-1}-y\leqslant -x$

$x\leqslant y\wedge y\leqslant x\leqslant^{-1}x=y$

$x\leqslant y\leqslant^{-1}x+z\leqslant y+z$

$x\leqslant y\wedge u\leqslant v\leqslant^{-1}x+u\leqslant y+v$

3.5 Contradictory propositions

Of two propositions, if one is true, the other is bound to be false, and if one is false, the other is bound to be true, then they are contradictory propositions. Suppose $p(x)$ and $q(x)$ are not distinguished propositions. $P(x)\leqslant^{-1}q(x)$ and $p(x)/\wedge^{-1}\neg q(x)$ are contradictory propositions. If the minset $(\langle p(x), \mathrm{T}\rangle\langle q(x), \mathrm{F}\rangle)$ is empty, then $p(x)\leqslant^{-1}q(x)$ is true and $p(x)/\wedge^{-1}\neg q(x)$ is false; otherwise, $p(x)\leqslant^{-1}q(x)$ is false and $p(x)/\wedge^{-1}\neg q(x)$ is true.

3.6 First-level decomposition

Decomposition includes hypothetical inference and disjunctive inference.

First-level hypothetical inference is from man(x)$\leqslant^{-1}$mortal(x) being true and man(Xunwei Zhou) being true to infer mortal(Xunwei Zhou) being true. From Section 2.4 we learn that a zeroth-order fact proposition can be obtained in two ways: being felt oneself, being inferred by first-level decomposition. Suppose we have truthified the major premise man(x)$\leqslant^{-1}$mortal(x). After we truthify the minor premise man(Xunwei Zhou), before we use Table 1.3 to infer mortal(Xunwei Zhou) to be true and feel ourselves mortal(Xunwei Zhou) to be true (when writing this book, Xunwei Zhou is still alive, you cannot feel mortal(Xunwei Zhou) to be true), we have not determined mortal(Xunwei Zhou) to be true. This is to say, the minor premise man(Xunwei Zhou) alone cannot decide the conclusion mortal(Xunwei Zhou). This property is called the first-level non-decidedness of minor premise to conclusion. If this property holds, then from the major premise man(x)$\leqslant^{-1}$mortal(x) and minor premise man(Xunwei Zhou) by Table 1.3 to infer the conclusion mortal(Xunwei Zhou) is from the known to the unknown, and the conclusion is a new knowledge to the minor premise.

Suppose we have truthified publish(x)$\vee/^{-1}$perish(x) and falsified publish(John), then by Table 2.14 we can infer perish(John).

3.7 Quasi-logical connection propositions

A quasi-logical connection proposition with the empirical or mathematical connection operator being $=^{-1}$ must stisfy at least one of the three conditions:

(1) Namesake fact proposition variables occur on both sides of the empirical or mathematical connection operator $=^{-1}$; e.g., $P=^{-1}P$, $P\vee Q=^{-1}Q\vee P$. These fact proposition variables are called quasi-relevantly bound fact proposition variables.

(2) Namesake fact proposition variables occur on one side of the empirical or mathematical connection operator $=^{-1}$ but on both sides of a binary fact composition operator; e.g., $P\vee\neg P=^{-1}\mathrm{U}_0$, $P\wedge\neg P=^{-1}\varnothing_0$. These fact proposition variables are called quasi-intermediarily bound fact proposition variables.

(3) Quasi-relevantly bound zeroth-level distinguished set $\varnothing_0$ or U_0 and a free fact proposition variable occur; $P\wedge\varnothing_0=^{-1}\varnothing_0$.

A quasi-logical connection proposition with the empirical or mathematical connection operator being $\leqslant^{-1}$ must satisfy the two conditions:

(1) It must be a meaningful proposition; e.g., $P\wedge Q\leqslant^{-1}P$, $P\leqslant^{-1}P\vee Q$ are meaningful propositions, while $P\wedge\neg P\leqslant^{-1}Q$ is a meaningless proposition.

(2) It must be a quasi-relevantly bound proposition; i.e., on both sides of $\leqslant^{-1}$ occur at least one namesake fact proposition variable that is called a quasi-relevantly bound fact proposition variable. For example, $P\wedge Q\leqslant^{-1}P$, $P\leqslant^{-1}P\vee Q$ are quasi-relevantly bound propositions, while $P\wedge\neg P\leqslant^{-1}Q$ is not.

The idea of quasi-relevant boundness comes from relevance logic where the necessary condition of A relevantly implying B is that A and B share the same propositional variable.

A true quasi-logical connection proposition is called a quasi-logical theorem, a false one a quasi-logical countertheorem.

The conjunction of some fact proposition variables or their negations is called an elementary product. For example, given fact proposition variables P and Q, then $\neg P\wedge Q$, $P\wedge\neg P$, $\neg Q\wedge P\wedge Q$ are elementary products.

Given an elementary product with n fact proposition variables, if for every fact proposition variable and its negation, at least one, and at most one, occurs in it, then it is called a minterm; e.g., given P and Q, $P\wedge Q$, $\neg P\wedge Q$, $P\wedge\neg Q$, $\neg P\wedge\neg Q$ are minterms. (Note the difference between minterms and minsets.)

A meaningful and bound quasi-logical connection proposition with the empirical or mathematical connection operator being $\leqslant^{-1}$ can be denoted by $\mathrm{A}(P_1, \ldots, P_j)\leqslant^{-1}\mathrm{B}(P_i, \ldots, P_k)$, $\mathrm{A}\leqslant^{-1}\mathrm{B}$ for short. Table 1.2 depicts its establishment by implicit inductive composition, as follows: first, a mutually inverse diagram containing all the minterms of the fact proposition variables $P_1, \ldots, P_j, P_i, \ldots, P_k$ is drawn, then whether the following two criteria are met or not is investigated:

(1) The minset (<A, T><B, F>) that constitutes the opposition area of $\mathrm{A}\leqslant^{-1}\mathrm{B}$ is empty;

(2) None of the minsets (<A, F><B, F>) and (<A, T><B, T>) that constitute the support area of $\mathrm{A}\leqslant^{-1}\mathrm{B}$ is empty.

If both (1) and (2) are met, then $A \leqslant^{-1} B$ is a quasi-logical theorem; otherwise, it is not a quasi-logical connection proposition.

A quasi-logical connection proposition with the empirical or mathematical connection operator being $=^{-1}$ can be denoted by $A(P_l, \ldots, P_j, \varnothing_0, U_0) =^{-1} B(P_i, \ldots, P_k, \varnothing_0, U_0)$, $A =^{-1} B$ for short. Table 2.7 depicts its establishment by implicit inductive composition, as follows: first, a mutually inverse diagram containing all the minterms of the fact proposition variables $P_l, \ldots, P_j, P_i, \ldots, P_k$ is drawn, then whether the following two criteria are met or not is investigated:

(3) The minsets (<A, T><B, F>) and (<A, F><B, T>) that constitute the opposition area of $A =^{-1} B$ are all empty;

(4) The minsets (<A, F><B, F>) and (<A, T><B, T>) that constitute the support area of $A =^{-1} B$ are all empty.

If (3) is met, then $A =^{-1} B$ is a quasi-logical theorem. If (4) is met, then $A =^{-1} B$ is a quasi-logical countertheorem. If none of (3) and (4) is met, then $A =^{-1} B$ is not a quasi-logical connection proposition.

Example 3.7: Truthify $P \wedge Q \leqslant^{-1} P$ by implicit inductive composition.

Proof: $P \wedge Q$ and P are neither empty sets nor universal sets, therefore, $P \wedge Q \leqslant^{-1} P$ is meaningful. P occurs on both sides of $\leqslant^{-1}$, therefore, $P \wedge Q \leqslant^{-1} P$ is quasi-relevantly bound. Inductive composition can go on. The mutually inverse diagram containing all the fact proposition variables P and Q is shown in Fig. 3.3.

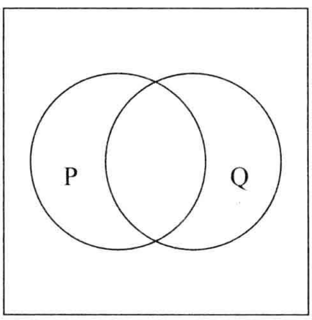


Fig. 3.3 Mutually inverse diagram for $P \wedge Q \leqslant^{-1} P$

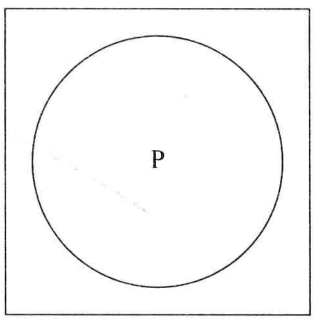


Fig. 3.4 Mutually inverse diagram for $P \wedge \varnothing_0 =^{-1} \varnothing_0$

In Fig. 3.3 there are 4 minterms. Minterm $P \wedge Q$ belongs to the minset (<$P \wedge Q$, T><P, T>) that constitutes the support area of $P \wedge Q \leqslant^{-1} P$. Minterms $\neg P \wedge Q$ and $\neg P \wedge \neg Q$ belong to the minset (<$P \wedge Q$, F><P, F>) that constitute the support area of $P \wedge Q \leqslant^{-1} P$; i.e., none of the two minsets that constitute the support area of $P \wedge Q \leqslant^{-1} P$ is empty. Thus, criterion (2) is met. Minterm $P \wedge \neg Q$ belongs to the minset (<$P \wedge Q$, F><P, T>) that constitutes the neutral area of $P \wedge Q \leqslant^{-1} P$. No minterm belongs to the minset (<$P \wedge Q$, T><P, F>) that

constitute the opposition area of $P\wedge Q\leqslant^{-1}P$; thus, criterion (1) is also met. Therefore, $P\wedge Q\leqslant^{-1}P$ is a quasi-logical theorem.

Example 3.8: Truthify $P\wedge\varnothing_0=^{-1}\varnothing_0$ by implicit inductive composition.

Proof: Let us draw the mutually inverse diagram containing all the minterms of the fact proposition variable P, shown in Fig. 3.4. In Fig.3.4, there are 2 minterms, P and $\neg P$, belonging to the minset $(\langle P\wedge\varnothing_0, \mathrm{F}\rangle\langle\varnothing_0, \mathrm{F}\rangle)$ that constitutes the support area of $P\wedge\varnothing_0=^{-1}\varnothing_0$; thus, criterion (3) is met. Therefore, $P\wedge\varnothing_0=^{-1}\varnothing_0$ is a quasi-logical theorem.

The following are quasi-logical theorems:

$P=^{-1}P$	law of identity
$\neg\neg P=^{-1}P$	double negation law
$P\wedge\varnothing_0=^{-1}\varnothing_0$	zero-one law
$P\vee\varnothing_0=^{-1}P$	zero-one law
$P\wedge \mathrm{U}_0=^{-1}P$	zero-one law
$P\vee \mathrm{U}_0=^{-1}\mathrm{U}_0$	zero-one law
$P\wedge\neg P=^{-1}\varnothing_0$	law of non-contradiction
$P\vee\neg P=^{-1}\mathrm{U}_0$	law of excluded middle
$P\wedge P=^{-1}P$	idempotent law
$P\vee P=^{-1}P$	idempotent law
$P\wedge Q=^{-1}Q\wedge P$	commutative law
$P\vee Q=^{-1}Q\vee P$	commutative law
$P\vee P\wedge Q=^{-1}P$	absorption law
$P\wedge\{P\vee Q\}=^{-1}P$	absorption law
$\neg\{P\wedge Q\}=^{-1}\neg P\vee\neg Q$	quasi-De Morgan's law
$\neg\{P\vee Q\}=^{-1}\neg P\wedge\neg Q$	quasi-De Morgan's law
$P\wedge\{Q\wedge R\}=^{-1}P\wedge Q\wedge R$	associative law
$P\vee\{Q\vee R\}=^{-1}P\vee Q\vee R$	associative law
$P\wedge\{Q\vee R\}=^{-1}P\wedge Q\vee P\wedge R$	distributive law
$P\vee Q\wedge R=^{-1}\{P\vee Q\}\wedge\{P\vee R\}$	distributive law
$P\leqslant^{-1}P$	
$P\wedge Q\leqslant^{-1}P$	
$P\leqslant^{-1}P\vee Q$	

The instances of the quasi-logical theorems with the empirical or mathematical connection operator being $\leqslant^{-1}$ do not have the property of first-level non-decidedness of minor premise to conclusion. For example, man(x)$\wedge$breathe(x)$\leqslant^{-1}$man(x), an instance of $P\wedge Q\leqslant^{-1}P$, does not have first-level non-decidedness of minor premise to conclusion, for when the minor premise man(x)$\wedge$breathe(x) is true, the conclusion man(x) is also true, no need to make hypothetical inference.

3.8 The decomposition system of first-level single quasi-predicate calculus

The decomposition systems of mutually-inversistic logic are based on decomposition. It is from the known to the unknown. It starts from the known propositions, and through the rules of inference, makes arguments, until the unknown proposition becomes a known one. The known corresponds to the premises in classical logic, the unknown proposition to the conclusion. In both logics, if the argument follows the accepted rules of inference, then it is called a valid argument. The decomposition systems of mutually-inversistic logic differs from the proving systems in classical logic in that in the latter, we are concerned only with the validity of an argument; the truth values of the premises and conclusion play no part in it. In the former, we are concerned not only with the validity of an argument but also with the truth values of the known propositions and the unknown one. An argument begins only if the known propositions are all true; the unknown proposition becomes a true one only if the argument is valid. Our purpose is to prove the unknown proposition to be true from the true known propositions through the valid argument.

In the decomposition systems of mutually-inversistic logic, there are the following grounds and rules of inference:

P ground: A known proposition may be introduced at any point in an argument.

T ground: A proposition may be introduced into an argument if it can be inferred by the preceding propositions in the argument.

Rule of conjunction: If both A and B are true, then $A \wedge B$ may be introduced into an argument.

Rule of change: $\neg A \vee \neg B$ being true may be changed into $\neg A \vee /^{-1} \neg B$ being true.

Rules of decomposition: All truth tables of decomposition of the connection operators are rules of inference. They amount to the three rules of inference: the affirmative expression of hypothetical inference, the negative expression of hypothetical inference, the disjunctive inference.

One proof method is called proof by contradiction (or indirect method of proof) which is performed as follows: If we want to prove the unknown proposition C, then we assume $\neg C$ to be true and reason backward, until contradiction occurs; thus, we prove C to be true.

In the decomposition system of first-level single quasi-predicate calculus, the propositions involved include fact propositions, single empirical or mathematical connection propositions, and quasi-logical connection propositions.

Example 3.9: Make argument according to the known: If x is y's parent, then x is y's ancestor; if x is y's parent and y is z's ancestor, then x is z's ancestor; Bob is Sam's parent; Sam is Ted's parent; The unknown: Bob is Ted's ancestor.

Solution:

The known:

$\langle \text{parent}(x, y) \leqslant^{-1} \text{ancestor}(x, y), \text{T} \rangle$

$\langle \text{parent}(x, y) \wedge \text{ancestor}(y, z) \leqslant^{-1} \text{ancestor}(x, z), \text{T} \rangle$

$\langle \neg\{P \wedge Q\} =^{-1} \neg P \vee \neg Q, \text{T} \rangle$

$\langle P \wedge \neg P =^{-1} \varnothing_0, \text{T} \rangle$

$\langle \text{parent(Bob, Sam)}, \text{T} \rangle$

$\langle \text{parent(Sam, Ted)}, \text{T} \rangle$

The unknown:

ancestor(Bob, Ted).

We adopt the indirect method of proof.

Proof:

(1) $\langle \neg \text{ancestor(Bob, Ted)}, \text{T} \rangle$	P (assumed)
(2) $\langle \text{parent}(x, y) \wedge \text{ancestor}(y, z) \leqslant^{-1} \text{ancestor}(x, z), \text{T} \rangle$	P
(3) $\langle \neg\{\text{parent(Bob}, y) \wedge \text{ancestor}(y, \text{Ted})\}, \text{T} \rangle$	T(1)(2)Table 1.4
(4) $\langle \neg\{P \wedge Q\} =^{-1} \neg P \vee \neg Q, \text{T} \rangle$	P
(5) $\langle \neg \text{parent(Bob}, y) \vee \neg \text{ancestor}(y, \text{Ted}), \text{T} \rangle$	T(3)(4)Table 2.8
(6) $\langle \neg \text{parent(Bob}, y) \vee/^{-1} \neg \text{ancestor}(y, \text{Ted}), \text{T} \rangle$	T(5)Rule of change
(7) $\langle \text{parent(Bob, Sam)}, \text{T} \rangle$	P
(8) $\langle \neg \text{ancestor(Sam, Ted)}, \text{T} \rangle$	T(6)(7)Table 2.14
(9) $\langle \text{parent}(x, y) \leqslant^{-1} \text{ancestor}(x, y), \text{T} \rangle$	P
(10) $\langle \neg \text{parent(Sam, Ted)}, \text{T} \rangle$	T(8)(9)Table 1.4
(11) $\langle \text{parent(Sam, Ted)}, \text{T} \rangle$	P
(12) $\langle \text{parent(Sam, Ted)} \wedge \neg \text{parent(Sam, Ted)}, \text{T} \rangle$	T(10)(11)Rule of conjunction
(13) $\langle P \wedge \neg P =^{-1} \varnothing_0, \text{T} \rangle$	P
(14) $\langle \varnothing_0, \text{T} \rangle$	T(12)(13)Table 2.8

From (14) we observe that the permanently false fact proposition $\varnothing_0$ has the value T; therefore, contradiction results, and ancestor(Bob, Ted) is proved.

The truth values of (1) to (14) are all T, which can be omitted.

Chapter 4 Second-level single quasi-predicate calculus

Second-level single quasi-predicate calculus studies the relationship between single empirical or mathematical connection propositions and single logical connection propositions, quasi-transcendent logical connection propositions.

4.1 Meaningless and meaningful second-order single logical connection propositions

$\{P=^{-1}Q\}\wedge\{P<^{-1}Q\}\leqslant^{-1}\{P\leqslant^{-1}Q\}$ is a meaningless second-order single logical connection proposition, for $\{P=^{-1}Q\}\wedge\{P<^{-1}Q\}$ is an empty set, a distinguished proposition. While $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is a meaningful second-order single logical connection proposition, for none of $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ and $P/\wedge^{-1}R$ is a distinguished proposition.

4.2 Free and bound second-order single logical connection propositions

Unlike the second-order predicate calculus of classical logic, the second-level single quasi-predicate calculus of mutually-inversistic logic does not quantify predicate and function variables.

A single logical connection proposition is a connection between two single empirical or mathematical connection propositions, and is connected by the same fact propositions. For example, there exists a connection between $P\leqslant^{-1}Q$ and $\neg Q\leqslant^{-1}\neg P$ in $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$, because they are connected by the same fact proposition variables P and Q. While there is no connection between $P\leqslant^{-1}Q$ and $\neg S\leqslant^{-1}\neg R$ in $\{P\leqslant^{-1}Q\}=^{-1}\{\neg S\leqslant^{-1}\neg R\}$, because there is not the same fact proposition variable to connect them. Namesake fact proposition variables occurring on both sides of a logical connection operator are called relevantly bound fact proposition variables; e.g., P and Q in $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ are relevantly bound fact proposition variables.

Sometimes, even though there are the same fact propositions between two single empirical or mathematical connection propositions, there is still no connection between them. For example, there is no connection between $\{P\leqslant^{-1}Q\}\wedge\{S\leqslant^{-1}R\}$ and $P\leqslant^{-1}R$ in $\{P\leqslant^{-1}Q\}\wedge\{S\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$, even though there are the same fact proposition variables, P

and R, between them. While there exists a connection between $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ and $P\leqslant^{-1}R$ in $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$. This is not only because there are the same fact proposition variables, P and R, to establish a connection between them but also because there is Q to act as an intermediary role. The fact proposition variable Q in $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is called an intermediarily bound fact proposition variable. A fact proposition variable is an intermediarily bound variable if it occurs on one side of a logical connection operator but on both sides of a binary empirical or mathematical composition operator.

If a fact proposition variable occurs on one side of a logical connection operator but on both sides of an empirical or mathematical connection operator, then it is an additionally bound variable. The idea comes from the inference "adding concept to judgment" in Aristotelian logic. For example, R in $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P\wedge R\leqslant^{-1}Q\wedge R\}$ is an additionally bound fact proposition variable.

If a fact proposition variable occurs on one side of an empirical or mathematical connection operator but on both sides of a fact composition operator, then it is a juxtaposed bound variable. For example, the first and second occurrences of $\{P\wedge P\leqslant^{-1}Q\}\leqslant^{-1}\{P\leqslant^{-1}Q\}$ is a juxtaposed bound fact proposition variable.

If a fact proposition variable occurs in a second-order single logical connection proposition only once, then it is a free variable. For example, Q and R in $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ are free fact proposition variables. If a fact proposition variable occurs in a second-order single logical connection proposition more than once, then it is a bound variable, for it can be at least one of the above mentioned boundness. For example, the first and second occurrences of P in $\{P\leqslant^{-1}Q\}\wedge\{P\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}Q\wedge R\}$ are intermediarily bound, the first and third occurrences of P are relevantly bound, the second and third occurrences of P are relevantly bound.

If all the fact proposition variables in a second-order single logical connection proposition are bound, then the proposition is a bound one. If at least one fact proposition variable is free, then the proposition is a free one.

If a second-order single logical connection proposition is meaningful and bound, then establishment by inductive composition of it can be made.

4.3 Second-level explicit inductive composition

Example 4.1: Truthify $\{\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ by explicit inductive composition.

Solution: $P\leqslant^{-1}Q$ and $P/\wedge^{-1}Q$ are neither empty sets nor universal sets, therefore, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is meaningful. P and Q are relevantly bound, therefore, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is bound. Inductive composition can go on. When P is assigned $x=y$, Q $x\leqslant y$, $\{P$

$\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ becomes $\{x=y\leqslant^{-1}x\leqslant y\}\leqslant^{-1}\{x=y/\wedge^{-1}x\leqslant y\}$, which holds. When P is assigned parent(x, y), Q ancestor(x, y), $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ becomes $\{$parent$(x, y)\leqslant^{-1}$ ancestor$(x, y)\}\leqslant^{-1}\{$parent$(x, y)/\wedge^{-1}$ancestor$(x, y)\}$, which also holds. Thus, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is truthified by explicit inductive composition.

Example 4.2: Falsify $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ by explicit inductive composition.

Solution: $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}$ and $P/\wedge^{-1}R$ are neither empty sets nor universal sets, therefore, $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is meaningful. P and R are relevantly bound fact proposition variables, Q is an intermediarily bound fact proposition variable, therefore, $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is bound. Inductive composition can go on. When P is assigned $x<y$, Q $x\leqslant y$, R $x=y$, $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ becomes $\{x<y\leqslant^{-1}x\leqslant y\}\wedge\{x\leqslant y/\wedge^{-1}x=y\}\leqslant^{-1}\{x<y/\wedge^{-1}x=y\}$. The antecedent $\{x<y\leqslant^{-1}x\leqslant y\}\wedge\{x\leqslant y/\wedge^{-1}x=y\}$ is true, the consequent $x<y/\wedge^{-1}x=y$ is false. Therefore, the assignment serves as a counterexample, falsifying $\{P\leqslant^{-1}Q\}\wedge\{Q/\wedge^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$.

4.4 Second-level implicit inductive composition

Suppose $\mathrm{A}\leqslant^{-1}\mathrm{B}$ is a meaningful and bound second-order single logical connection proposition; its establishment by implicit inductive composition goes like this: In the term space, we draw mutually inverse diagrams of A. If they are all mutually inverse diagrams of B, then $\mathrm{A}\leqslant^{-1}\mathrm{B}$ is truthified by implicit inductive composition; if at least one is not a mutually inverse diagram of B, then $\mathrm{A}\leqslant^{-1}\mathrm{B}$ is falsified by implicit inductive composition.

Example 4.3: Truthify $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ by implicit inductive composition.

Proof: $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ and $P\leqslant^{-1}R$ are neither empty sets nor universal sets, therefore, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is meaningful. P and R are relevantly bound fact proposition variables, Q is an intermediarily bound fact proposition variable, therefore, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is bound. Inductive composition can go on.

The mutually inverse diagrams of $P\leqslant^{-1}Q$ are shown in Fig. 4.1 (1) and (2), those of $Q\leqslant^{-1}R$ are shown in Fig. 4.1 (3) and (4). The mutually inverse diagrams of $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$; i.e., Fig. 4.1 (5) to (8), are obtained by drawing each mutually inverse diagram of $P\leqslant^{-1}Q$ together with each of $Q\leqslant^{-1}R$. This can be done because the Q appearing in $P\leqslant^{-1}Q$ and the Q in $Q\leqslant^{-1}R$ are different occurrences of the same fact proposition variable. Take Fig. 4.1 (6) as an example. It is obtained by drawing Fig. 4.1 (1) and (4) together. Considering the intermediary role played by Q, we establish a unicellular relationship between P and R; i.e., $P<^{-1}R$, which is part of the consequent $P\leqslant^{-1}R$. Likewise, Fig. 4. (5), (7), and (8) are also mutually inverse diagrams of $P\leqslant^{-1}R$. Thus, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is 47

truthified by implicit inductive composition.

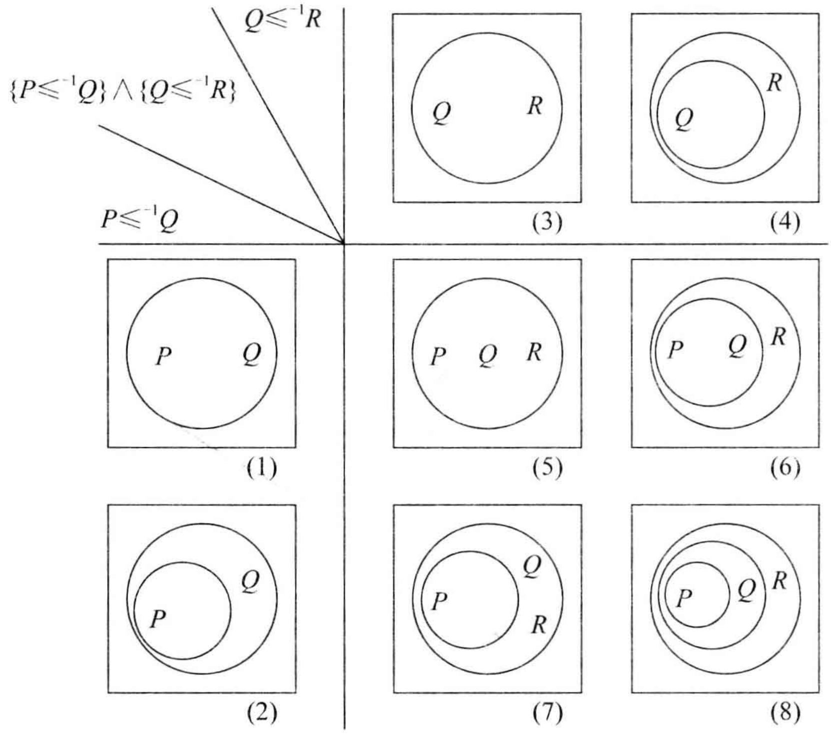


Fig. 4.1 Mutually inverse diagram for $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$

Example 4.4: Falsify $\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$ by implicit inductive composition.

Disproof: $P\wedge Q\leqslant^{-1}R$ and $\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$ are neither empty sets nor universal sets, therefore, $\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$ is meaningful. P, Q, and R are all bound fact proposition variables, therefore, $\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$ is bound. Inductive composition can go on. One of the mutually inverse diagrams of $P\wedge Q\leqslant^{-1}R$ is shown in Fig. 4.2 in which neither $P\leqslant^{-1}R$ nor $Q\leqslant^{-1}R$ holes; thus, $\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$ is falsified by implicit inductive composition.

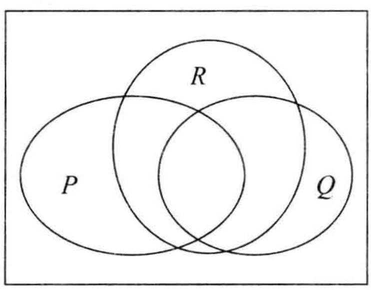


Fig. 4.2 Mutually inverse diagram for $\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}$

Single logical theorems include but are not limited to the following:

$P\vee/^{-1}Q=^{-1}Q\vee/^{-1}P$	commutative law
$P/\wedge^{-1}Q=^{-1}Q/\wedge^{-1}P$	commutative law
$\neg\{P/\wedge^{-1}Q\}=^{-1}\neg P\vee/^{-1}\neg Q$	single De Morgan's law
$\neg\{P\vee/^{-1}Q\}=^{-1}\neg P/\wedge^{-1}\neg Q$	single De Morgan's law
$\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$	contrapositive law
$\{P\leqslant^{-1}Q\}=^{-1}\{\neg P\vee/^{-1}Q\}$	
$\neg\{P\leqslant^{-1}Q\}=^{-1}P/\wedge^{-1}\neg Q$	
$\neg\{P/\wedge^{-1}Q\}=^{-1}P\leqslant^{-1}\neg Q$	
$\{P\leqslant^{-1}Q\}=^{-1}\{P/\wedge^{-1}Q\}$	
$\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P\wedge R\leqslant^{-1}Q\wedge R\}$	
$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}P\}=^{-1}\{P=^{-1}Q\}$	
$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$	
$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$	
$\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$	
$\{P\leqslant^{-1}R\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\vee Q\leqslant^{-1}R\}$	
$\{P\leqslant^{-1}R\}\vee\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\wedge Q\leqslant^{-1}R\}$	
$\{P\leqslant^{-1}Q\}\wedge\{P\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}Q\wedge R\}$	
$\{P\leqslant^{-1}Q\}\vee\{P\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}Q\vee R\}$	
$\{P\leqslant^{-1}Q\}\wedge\{\neg P\leqslant^{-1}R\}\leqslant^{-1}\{Q\vee/^{-1}R\}$	
$\{P\leqslant^{-1}\neg R\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{\neg P\vee/^{-1}\neg Q\}$	
$\{P\leqslant^{-1}Q\}\wedge\{P\leqslant^{-1}R\}\leqslant^{-1}\{Q/\wedge^{-1}R\}$	
$\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\wedge\neg R\leqslant^{-1}\neg Q\}$	
$\{P\wedge\neg Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}Q\vee R\}$	
$\{P\leqslant^{-1}Q\}\wedge\{R\leqslant^{-1}S\}\wedge\{P/\wedge^{-1}R\}\leqslant^{-1}\{Q/\wedge^{-1}S\}$	constructive dilemma
$\{P\leqslant^{-1}Q\}\wedge\{R\leqslant^{-1}S\}\wedge\{P\vee/^{-1}R\}\leqslant^{-1}\{Q\vee/^{-1}S\}$	constructive dilemma
$\{P\leqslant^{-1}Q\}\wedge\{R\leqslant^{-1}S\}\wedge\{\neg Q\vee/^{-1}\neg S\}\leqslant^{-1}\{\neg P\vee/^{-1}\neg R\}$	destructive dilemma

4.5 Second-level decomposition

Second-level hypothetical inference is from $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ being true and integer(x)$\leqslant^{-1}$rational(x) being true to infer $\neg$rational(x)$\leqslant^{-1}\neg$integer(x) being true.

From Section 2.4 we see that a first-order empirical or mathematical connection proposition can be obtained in two ways: by second-level decomposition and by first-level inductive composition. Suppose we have truthified $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$. After we truthify integer(x)$\leqslant^{-1}$rational(x), before we use second-level decomposition and first-level inductive composition, we have not determined $\neg$rational(x)$\leqslant^{-1}\neg$integer(x) to be true. This is to

say that the minor premise integer(x)$\leqslant^{-1}$rational(x) alone cannot decide the conclusion $\neg$rational(x)$\leqslant^{-1}\neg$integer(x). This property is called the second-level non-decidedness of minor premise to conclusion. If this property holds, then from the major premise and the minor premise to infer the conclusion is from the known to the unknown, and the conclusion is a new knowledge to the minor premise; otherwise, the conclusion is not a new knowledge to the minor premise.

4.6 Quasi-transcendent logical connection propositions

Quasi-transcendent logical connection propositions are obtained by lifting Section 3.7 quasi-logical connection propositions up one level. By lifting up one level the author means that fact propositions P, Q, R, $\varnothing_0$, and U_0 are lifted to single empirical or mathematical connection propositions Ψ, Ω, Θ, $\varnothing_1$, and U_1; fact composition operators $\neg$, $\wedge$, and $\vee$ are lifted to empirical or mathematical composition operators $\neg$, $\wedge$, and $\vee$; empirical or mathematical connection operators $=^{-1}$ and $\leqslant^{-1}$ are lifted to logical connection operators $=^{-1}$ and $\leqslant^{-1}$. For example, after lifting up one level the quasi-De morgan's law $\neg\{P\wedge Q\}=^{-1}\neg P\vee\neg Q$ becomes quasi-transcendent De Morgan's law $\neg\{\Psi\wedge\Omega\}=^{-1}\neg\Psi\vee\neg\Omega$.

4.7 Knowledge-cognition science

In classical logic, a rule is both a tautology and an inference rule. For example, hypothetical syllogism is both a tautology $(P\rightarrow Q)\wedge(Q\rightarrow R)\Rightarrow P\rightarrow R$ and an inference rule:

$$\begin{array}{l} P\rightarrow Q \\ Q\rightarrow R \\ \hline \therefore P\rightarrow R. \end{array}$$

And the affirmative expression of hypothetical inference is both a tautology $P\wedge(P\rightarrow Q)\Rightarrow Q$ and an inference rule:

$$\begin{array}{l} P\rightarrow Q \\ P \\ \hline \therefore Q. \end{array}$$

A tautology in classical logic corresponds to a logical theorem in mutually-inversistic logic. In mutually-inversistic logic, a rule is either a logical theorem or an inference rule, but not both. For example, hypothetical syllogism is a logical theorem but not an inference rule, the affirmative expression of hypothetical inference is an inference rule but not a logical theorem. As mentioned in Section 3.8, in mutually-inversistic logic, there are only 5 inference rules: rule of conjunction, rule of change from $\vee$ to $\vee/^{-1}$, the affirmative expression

of hypothetical inference, the negative expression of hypothetical inference, the disjunctive inference. The other rules are all logical theorems.

A logical theorem is knowledge. While an inference rule is a cognition from the known knowledge to the unknown knowledge. An inference rule is represented by a horizontal line, above the line are the known, below the line is the unknown. Mutually-inversistic logic is a knowledge-cognition science, studying knowledge and cognition from the known (knowledge) to the unknown (knowledge). It distinguishes knowledge from cognition, while classical logic does not distinguish knowledge from cognition.

4.8 The decomposition system of second-level single quasi-predicate calculus

In the decomposition system of second-level single quasi-predicate calculus, the propositions involved include single empirical or mathematical connection propositions, single logical connection propositions, and quasi-transcendent logical connection propositions.

As mentioned in Section 3.8, in all the steps of an argument, the truth values are always T; so, we omit them.

Example 4.5: Make an argument according to the known: if a man fears difficulty then he cannot succeed, a man either succeeds or fails, some men do not fail. The unknown: some men do not fear difficulty.

Solution: suppose

man(x): x is a man,

fear_difficulty(x): x fears difficulty,

succeed(x): x succeeds,

fail(x): x fails

The known:

fear_difficulty(x)$\leqslant^{-1}\neg$succeed(x),

succeed(x)$\vee/^{-1}$ fail(x),

man(x)$/\wedge^{-1}\neg$fail(x),

$\{P\vee/^{-1}Q\}=^{-1}\{\neg P\leqslant^{-1}Q\}$,

$P\vee/^{-1}Q=^{-1}Q\vee/^{-1}P$,

$\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$,

$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$,

$\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$.

The unknown:

man(x) $/\wedge^{-1}\neg$fear_difficulty(x).

Proof:

(1) fear_difficulty(x)$\leqslant^{-1}\neg$succeed(x) P

(2) $\{P\leqslant^{-1}Q\}=^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ P

(3) succeed(x) $\leqslant^{-1}\neg$fear_difficulty(x) T (1) (2) Table 2.8

(4) succeed(x)$\vee/^{-1}$ fail(x) P

(5) $P\vee/^{-1}Q=^{-1}Q\vee/^{-1}P$ P

(6) fail(x)$\vee/^{-1}$ succeed(x) T (4) (5) Table2.8

(7) $\{P\vee/^{-1}Q\}=^{-1}\{\neg P\leqslant^{-1}Q\}$ P

(8) $\neg$fail(x)$\leqslant^{-1}$ succeed(x) T (6) (7) Table2.8

(9) $\{\neg$fail(x)$\leqslant^{-1}$ succeed(x)$\}\wedge\{$succeed(x)$\leqslant^{-1}\neg$fear_difficulty(x)$\}$

T (8) (3) Rule of conjunction

(10) $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ P

(11) $\neg$fail(x) $\leqslant^{-1}\neg$fear_difficulty(x) T (9) (10) Table1.3

(12) man(x)$/\wedge^{-1}\neg$fail(x) P

(13)$\{$ man(x)$/\wedge^{-1}\neg$fail(x)$\}\wedge\{\neg$fail(x)$\leqslant^{-1}\neg$fear_difficulty(x)$\}$

T (12) (11) Rule of conjunction

(14) $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ P

(15) man(x)$/\wedge^{-1}\neg$fear_difficulty(x) T (13) (14) Table1.3

Example 4.6: Make an argument according to the known: if x is an integer, then x is a rational; if x is a rational, then x is a real number. The unknown: some integers are real numbers.

Solution: Suppose

int(x): x is an integer,

rat(x): x is a rational,

real(x): x is a real number.

The known:

int(x)$\leqslant^{-1}$rat(x),

rat(x)$\leqslant^{-1}$real(x),

$\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$,

$\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$,

$\neg\{\Psi\wedge\Omega\}=^{-1}\neg\Psi\vee\neg\Omega$.

The unknown:

int(x)$/\wedge^{-1}$real(x).

Proof:

We adopt the indirect method of proof.

Proof:

(1) $\neg\{\mathrm{int}(x)/\wedge^{-1}\mathrm{real}(x)\}$	P(assumed)
(2) $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$	P
(3) $\neg\{\{\mathrm{int}(x)/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}\mathrm{real}(x)\}\}$	T (1) (2) Table 1.4
(4) $\neg\{\Psi\wedge\Omega\}=^{-1}\neg\Psi\vee\neg\Omega$	P
(5) $\neg\{\mathrm{int}(x)/\wedge^{-1}Q\}\vee\neg\{Q\leqslant^{-1}\mathrm{real}(x)\}$	T (3) (4) Table 2.8
(6) $\neg\{\mathrm{int}(x)/\wedge^{-1}Q\}\vee/^{-1}\neg\{Q\leqslant^{-1}\mathrm{real}(x)\}$	T (5)Rule of change
(7) $\mathrm{rat}(x)\leqslant^{-1}\mathrm{real}(x)$	P
(8) $\neg\{\mathrm{int}(x)/\wedge^{-1}\mathrm{rat}(x)\}$	T (6) (7) Table2.15
(9) $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$	P
(10) $\neg\{\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)\}$	T (8) (9) Table1.4
(11) $\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)$	P
(12) $\neg\{\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)\}\wedge\{\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)\}$	T (10) (11) Rule of conjunction

From (12) we observe that contradiction results, therefore $\mathrm{int}(x)/\wedge^{-1}\mathrm{real}(x)$ is proved.

Chapter 5
First-level multiple predicate calculus

First-level multiple predicate calculus studies the relationship between fact propositions and multiple empirical or mathematical connection propositions.We regard multiple empirical or mathematical connection propositions as the generalization of single empirical or mathematical connection propositions. First, therefore, we study the features of single empirical or mathematical connection propositions; we then generalize them to multiple empirical or mathematical connection propositions.

5.1 Property fact proposition segments vs. non-property fact propositions

In single empirical or mathematical connection propositions such as $p(x)\leqslant^{-1}q(x)$, $p(x)\leqslant^{-1}\neg q(x)$, $p(x)/\wedge^{-1}q(x)$, $p(x)/\wedge^{-1}\neg q(x)$, p's are called property predicates, $p(x)$ is called a property fact proposition, and $p(x)\leqslant^{-1}$ and $p(x)/\wedge^{-1}$ are called property fact proposition segments, which are part of property proposition segments. In a single empirical or mathematical connection proposition, a property proposition segment is made up of only one property fact proposition segment, while in a multiple empirical or mathematical connection proposition, it is made up of more than one.

In the above single empirical or mathematical connection propositions q's are called non-property predicates, $q(x)$ is called a positive non-property fact proposition, and $\neg q(x)$ a negative non-property fact proposition. Positive non-property fact proposition and negative non-property fact proposition are called by a joint name, non-property fact proposition. $P(x)\leqslant^{-1}q(x)$ and $p(x)/\wedge^{-1}q(x)$ are single empirical or mathematical connection propositions with positive non-property fact propositions. $P(x)\leqslant^{-1}\neg q(x)$ and $p(x)/\wedge^{-1}\neg q(x)$ are single empirical or mathematical connection propositions with negative non-property fact propositions.

5.2 Mutually inverse multiple diagrams

A multiple empirical or mathematical connection proposition is depicted by a mutually inverse multiple diagram, a single empirical or mathematical connection proposition by a mutually inverse diagram. A multiple empirical or mathematical connection proposition is the generalization of a single empirical or mathematical connection proposition. A mutually

inverse multiple diagram is also the generalization of a mutually inverse diagram. Now, we derive the mutually inverse multiple diagram from the mutually inverse diagram of a single empirical or mathematical connection proposition, study its features, and generalize them to a multiple empirical or mathematical connection proposition.

The three steps listed below describe the procedure to derive the mutually inverse multiple diagram from the mutually inverse diagram of a true first-order single empirical or mathematical connection proposition:

(1) Draw the mutually inverse diagram of the proposition;

(2) Investigate only the set corresponding to the property fact proposition in the mutually inverse diagram;

(3) In the set, if the non-property fact proposition is a positive one, then mark the zeroth-level elements in its corresponding set with "○", and mark the zeroth-level elements in its negation set with "×"; if the non-property fact proposition is a negative one, then mark the zeroth-level elements in its corresponding set with "×", and mark the zeroth-level elements in its negation set with "○".

Example 5.1: Suppose the universe of discourse of terms is composed of a stone, a cat, and three persons: Alva, Carol, and Dale. Proposition man(x)$\leqslant^{-1}$mortal(x) is true for the universe. The mutually inverse multiple diagram of man(x)$\leqslant^{-1}$mortal(x) is derived from its mutually inverse diagram, shown below:

Solution:

Step (1): Draw the mutually inverse diagram of man(x)$\leqslant^{-1}$mortal(x) shown in Fig. 5.1.

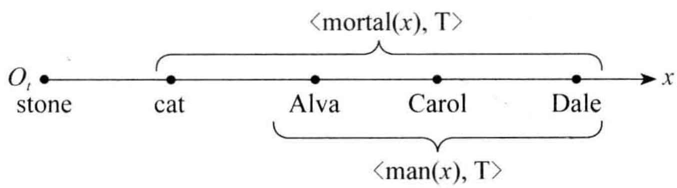


Fig. 5.1 Mutually inverse diagram for man(x)$\leqslant^{-1}$mortal(x)

Step (2): Consider only the set ⟨man(x), T⟩ corresponding to the property fact proposition man(x) (see Fig. 5.2).

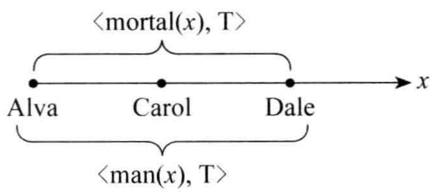


Fig. 5.2 Transition from mutually inverse diagram to mutually inverse multiple diagram

Step (3): Since the non-property fact proposition mortal(x) is a positive one, and the set ⟨mortal(x), T⟩ contains all the zeroth-level elements of ⟨man(x), T⟩, we mark them all with "O" (see Fig. 5.3).

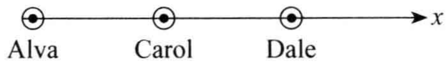


Fig. 5.3 Mutually inverse multiple diagram for man(x)$\leqslant^{-1}$mortal(x)

Example 5.2: The universe of discourse of terms is composed of a meteor, a stone, and three persons: Alva, Dale, and Armstrong. Proposition man(x)$/\wedge^{-1}\neg$land_on_Moon(x) is true for the universe. The mutually inverse multiple diagram of the proposition is derived from its mutually inverse diagram, shown below:

Solution:

Step (1): Draw the mutually inverse diagram of man(x)$/\wedge^{-1}\neg$land_on_Moon(x) shown in Fig. 5.4.

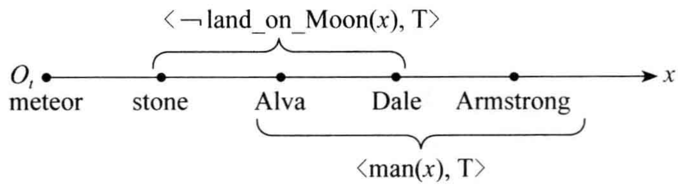


Fig. 5.4 Mutually inverse diagram for man(x)$/\wedge^{-1}\neg$land_on_Moon(x)

Step (2): Consider only the set ⟨man(x), T⟩ corresponding to the property fact proposition man(x) (see Fig. 5.5).

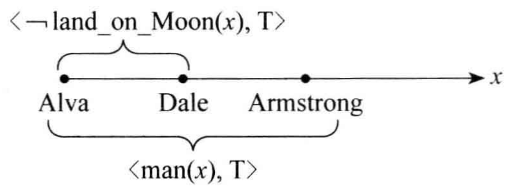


Fig. 5.5 Transition from mutually inverse diagram to mutually inverse multiple diagram

Step (3): The non-property fact proposition $\neg$land_on_Moon(x) is a negative one, and the set ⟨$\neg$land_on_Moon(x), T⟩ contains two zeroth-level elements: Alva and Dale; we mark them with "×". Its negation set ⟨$\neg$land_on_Moon(x), F⟩ contains only one element: Armstrong; we mark it with "o" (see Fig. 5.6).

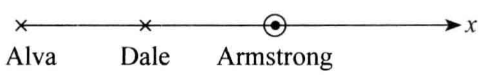


Fig. 5.6 Mutually inverse multiple diagram for man(x)$/\wedge^{-1}\neg$land_on_Moon(x)

Note that the mutually inverse multiple diagram of man(x)$/\wedge^{-1}$ land_on_Moon(x) is the same as Fig. 5.6.

5.3 Success diagrams vs. failure diagrams

In Section 5.2 we discussed the problem: given a true first-order single empirical or mathematical connection proposition, draw its mutually inverse multiple diagram. In this section we discuss its inverse problem: given a mutually inverse multiple diagram, determine whether a first-order single empirical or mathematical connection proposition is true or not according to it. This is a simplified version of establishment by explicit inductive composition of a first-order single empirical or mathematical connection proposition, equivalent to investigating only the right half of Fig. 2.6.

Given a mutually inverse multiple diagram, a first-order single empirical or mathematical connection proposition can be either true or false. If it is true, then the mutually inverse multiple diagram is said to be its success diagram; otherwise, the mutually inverse multiple diagram is said to be its failure diagram.

Example 5.3: Given the mutually inverse multiple diagram shown in Fig. 5.7, determine whether it is a success or failure diagram of the following first-order single empirical or mathematical connection propositions.

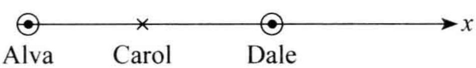


Fig. 5.7 Mutually inverse multiple diagram for Example 5.3

(1) man(x)$\leqslant^{-1}$bearded(x);

(2) man(x)$\leqslant^{-1}\neg$bearded(x);

(3) man(x)$/\wedge^{-1}$bearded(x);

(4) man(x)$/\wedge^{-1}\neg$bearded(x).

Solution: Fig. 5.7 is a failure diagram of man(x)$\leqslant^{-1}$bearded(x); a failure diagram of man(x)$\leqslant^{-1}\neg$bearded(x); a success diagram of man(x)$/\wedge^{-1}$bearded(x); a success diagram of man(x)$/\wedge^{-1}\neg$bearded(x).

5.4 Least success diagrams

The form a success diagram of a proposition at least should take is called the least success diagram of the proposition.

We take the first-order single empirical or mathematical connection propositions, man(x)$\leqslant^{-1}$bearded(x), man(x)$/\wedge^{-1}$bearded(x), man(x)$\leqslant^{-1}\neg$bearded(x), and man(x)$/\wedge^{-1}$

$\neg$bearded(x) as examples, to see what their least success diagrams look like. Suppose there are three zeroth-level elements: Alva, Carol, and Dale, in the set corresponding to the property fact proposition man(x), we can draw at most eight mutually inverse multiple diagrams, shown in Fig. 5.8 (1) to (8).

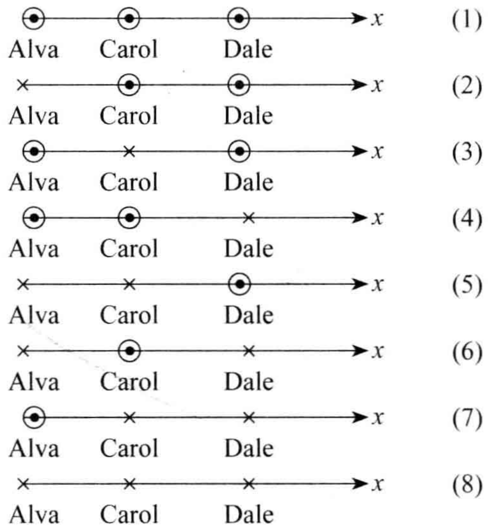


Fig. 5.8 Least success diagrams

In order for man(x)$\leqslant^{-1}$bearded(x) to be true, at least all people in the set ⟨man(x), T⟩ should be bearded, therefore Fig. 5.8 (1) is not only its sole success diagram but also its least success diagram. Fig. 5.8 (1) to (7) are all success diagrams for man(x)/$\wedge^{-1}$bearded(x). In order for man(x)/$\wedge^{-1}$bearded(x) to be true, at least one person in the set ⟨man(x), T⟩ should be bearded. Therefore Fig. 5.8 (5) to (7) are its least success diagrams. Likewise, Fig. 5.8 (8) is the least success diagram for man(x)$\leqslant^{-1}\neg$bearded(x), and Fig. 5.8 (2) to (4) are the least success diagrams for man(x)/$\wedge^{-1}\neg$bearded(x).

A second-order single empirical or mathematical connection proposition does not correspond to a concrete universe of discourse of terms, but its least success diagram can still be drawn. The least success diagram of $p(x)\leqslant^{-1}q(x)$ is shown in Fig. 5.9.

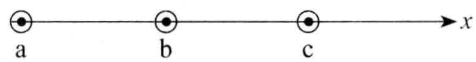


Fig.5.9 Least success diagram for $p(x)\leqslant^{-1}q(x)$

One of the least success diagrams of $p(x)/\wedge^{-1}q(x)$ is shown in Fig. 5.10, which can be simplified to Fig. 5.11.

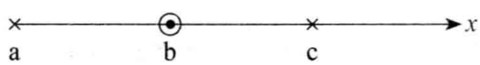


Fig. 5.10 Least success diagram for $p(x)/\wedge^{-1}q(x)$

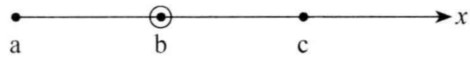


Fig. 5.11 Simplified least success diagram for $p(x)/\wedge^{-1}q(x)$

The least success diagram of $p(x)\leqslant^{-1}\neg q(x)$ is shown in Fig. 5.12.

Fig. 5.12 Least success diagram for $p(x)\leqslant^{-1}\neg q(x)$

One of the least success diagrams of $p(x)/\wedge^{-1}\neg q(x)$ is shown in Fig. 5.13, which can be simplified to Fig. 5.14.

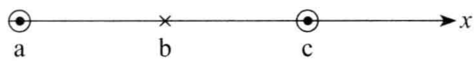


Fig. 5.13 Least success diagram for $p(x)/\wedge^{-1}\neg q(x)$

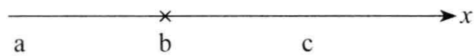


Fig. 5.14 Simplified least success diagram for $p(x)/\wedge^{-1}\neg q(x)$

The common characteristic of the universal propositions man(x)$\leqslant^{-1}$bearded(x) and man(x)$\leqslant^{-1}\neg$bearded(x) is that all zeroth-level elements in the property fact proposition man(x) satisfy the non-property fact proposition. We can merge the two universal propositions into property proposition segment man(x)$\leqslant^{-1}$, the least success diagram of which is shown in Fig. 5.15.

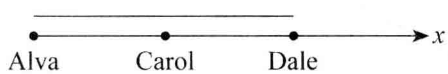


Fig. 5.15 Least success diagram for man(x)$\leqslant^{-1}$

The common characteristic of the existential propositions man(x)$/\wedge^{-1}$bearded(x) and man(x)$/\wedge^{-1}\neg$bearded(x) is that at least one zeroth-level element in the property fact proposition man(x) satisfies the non-property propositions. We can merge the two existential proposition into property proposition segment man(x)$/\wedge^{-1}$, one of the least success diagrams of which is shown in Fig. 5.16.

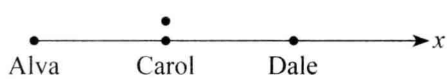


Fig. 5.16 Least success diagram for man(x)$/\wedge^{-1}$

The least success diagram of the property proposition segment $p(x)\leqslant^{-1}$ is shown in Fig. 5.17.

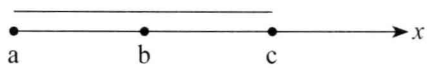


Fig. 5.17 Least success diagram for $p(\mathbf{x})\leqslant^{-1}$

One of the least success diagrams of the property proposition segment $p(x)/\wedge^{-1}$ is shown in Fig. 5.18.

Fig. 5.18 Least success diagram for $p(x)/\wedge^{-1}$

5.5 Proposition chains and property proposition segment chains

Single empirical or mathematical connection propositions with the same non-property fact proposition form a proposition chain. The proposition chain formed by the first-order single empirical or mathematical connection propositions with the positive non-property fact proposition is shown in Fig. 5.19 where a least success diagram for each proposition in the chain is given at the right of it.

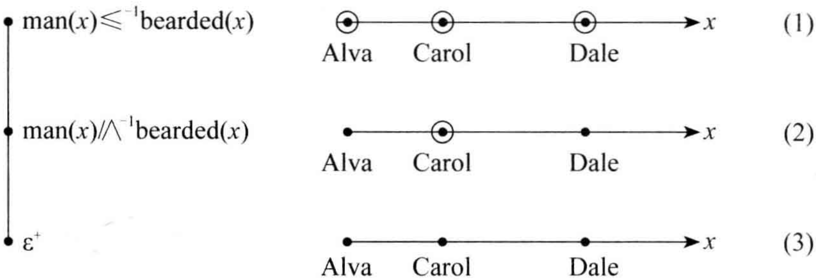


Fig.5.19 Proposition chain of the first-order single empirical or mathematical connection propositions with the positive non-property fact proposition

ε^{+} in the chain is called a positive null proposition, in which all mutually inverse multiple diagrams in Fig. 5.8 are its success diagrams and Fig. 5.8 (8), simplified as Fig. 5.19 (3), is its least success diagram.

The proposition chain formed by the first-order single empirical or mathematical connection propositions with the negative non-property fact proposition is shown in Fig. 5.20 where a least success diagram for each proposition in the chain is given at the right of it.

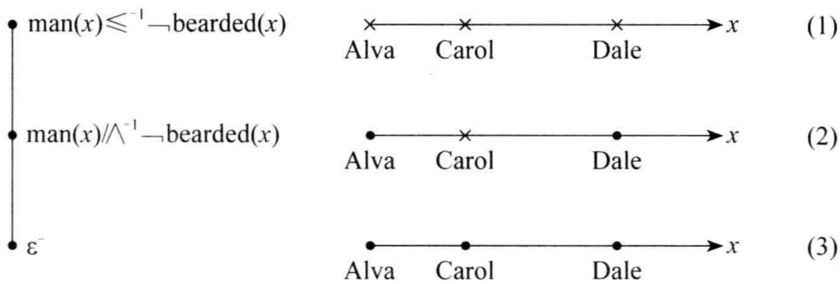


Fig. 5.20 Proposition chain of the first-order single empirical or mathematical connection propositions with the negative non-property fact proposition

ε^- in the chain is called a negative null proposition, in which all mutually inverse multiple diagrams in Fig. 5.8 are its success diagrams and Fig. 5.8 (1), simplified as Fig. 5.20 (3), is its least success diagram.

The proposition chain formed by the second-order single empirical or mathematical connection propositions with the positive non-property fact proposition is shown in Fig. 5.21.

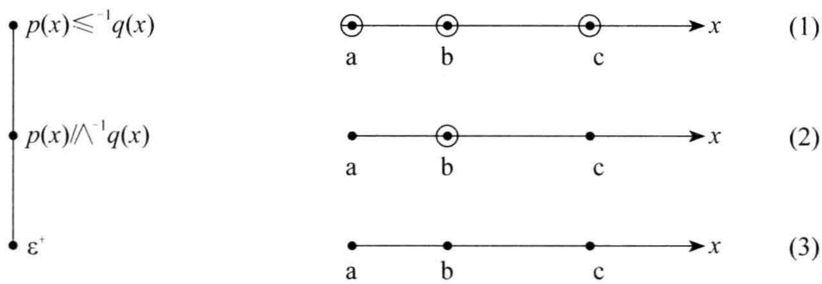


Fig. 5.21 Proposition chain of the second-order single empirical or mathematical connection propositions with the positive non-property fact proposition

The proposition chain formed by the second-order single empirical or mathematical connection propositions with the negative non-property fact proposition is shown in Fig. 5.22.

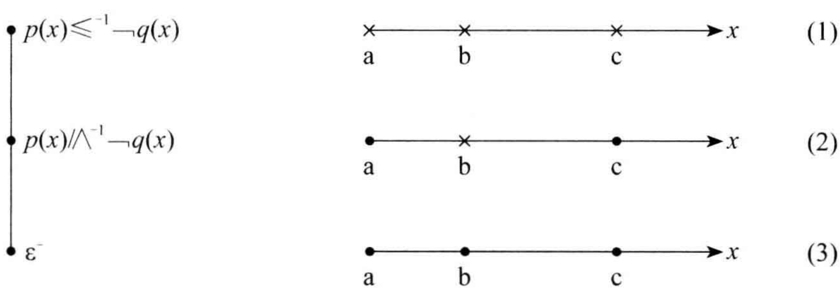


Fig. 5 22 Proposition chain of the second-order single empirical or mathematical connection propositions with the negative non-property fact proposition

In a chain, the proposition at the top is called the top proposition, the proposition at the bottom is called the bottom proposition, and the proposition second from the bottom is called the next to the bottom proposition. The top proposition and the bottom one have a unique least success diagram; others do not.

Of a pair of propositions in the chain, the upper one is called the upper proposition, and the lower one is called the lower proposition. Of a pair of adjacent propositions in the chain, the upper one is called the adjacent upper proposition, and the lower one is called the adjacent lower proposition. Any pair of propositions in a chain has the property that the upper proposition mutually inversely implies the lower proposition or that the success diagram of the upper proposition is bound to be one of the lower proposition while the success diagram of the lower proposition may not be one of the upper proposition; in other words, the least success diagram of the upper proposition must be a success diagram of the lower proposition, while the least success diagram of the lower proposition must not be a success diagram of the upper proposition.

The two proposition chains of Figs. 5.19 and 5.20 can be merged into a property proposition segment chain, shown in Fig. 5.23, where ε is the merger of ε^+ and ε^-.

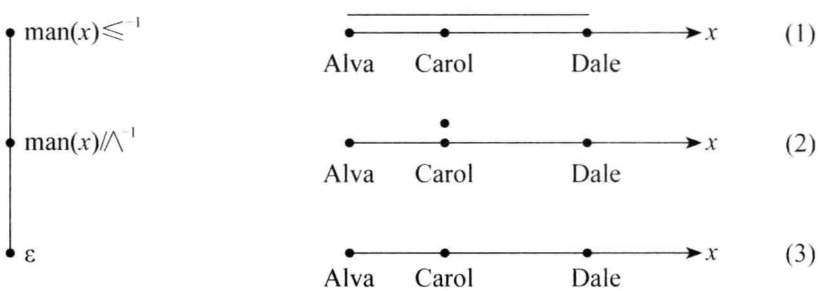


Fig. 5.23 Property proposition segment chain of first-order single empirical or mathematical connection propositions

The two proposition chains of Figs. 5.21 and 5.22 can be merged into a property proposition segment chain shown in Fig. 5.24.

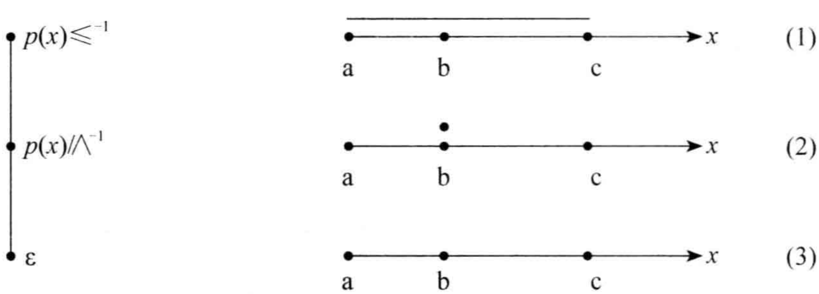


Fig. 5.24 Property proposition segment chain of second-order single empirical or mathematical connection propositions

5.6 Multiple empirical or mathematical connection propositions

"Behind every successful man there is a woman" is a first-order multiple empirical or mathematical connection proposition, denoted by:

$$\text{man}(x)\wedge\text{successful}(x)\leqslant^{-1}\text{woman}(y)/\wedge^{-1}\text{behind}(y, x) \quad (5.1)$$

"There exists th e least natural number" is a first-order multiple empirical or mathematical connection proposition, denoted by:

$$\text{natural\_number}(x)/\wedge^{-1}\{\text{natural\_number}(y)\leqslant^{-1}x\leqslant y\} \quad (5.2)$$

An example of a second-order multiple empirical or mathematical connection proposition includes:

$$p(x)\leqslant^{-1}q(y)/\wedge^{-1}r(x, y) \quad (5.3)$$

A multiple empirical or mathematical connection proposition is composed of a property proposition segment and a non-property proposition. Taking (5.2) as an example, its property proposition segment is natural_number(x)/$\wedge^{-1}${natural_number(y)$\leqslant^{-1}$, and its non-property proposition is $x\leqslant y$. In this book, we study only the simplest multiple empirical or mathematical connection propositions, such as (5.2) and (5.3), whose features are that the property proposition segment is composed of n (n>1) property fact propositions, each of which is a one-place predicate, that the non-property proposition is an n-place predicate.

5.7 Meaningful and bound multiple empirical or mathematical connection propositions

A multiple empirical or mathematical connection proposition is a meaningless one if at least one fact proposition in it is a distinguished fact proposition, or one property predicate is the negation of another. For example, natural_number(x) $\leqslant^{-1}\neg$natural_number(x)/$\wedge^{-1}$ $x\leqslant y$ is a meaningless first-order multiple empirical or mathematical connection proposition because the conjunction set of natural_number(x) and $\neg$natural_number(x) is an empty set. If a multiple empirical or mathematical connection proposition is not meaningless, then it is meaningful; e.g., lock(x)$\leqslant^{-1}$key(y)/$\wedge^{-1}$unlock(y, x) is a meaningful first-order multiple empirical or mathematical connection proposition.

A multiple empirical or mathematical connection proposition is a free one if at least one term variable in it is free. For example, lock(x)$\leqslant^{-1}$key(y)/$\wedge^{-1}$unlock(y, z) is a free first-order multiple empirical or mathematical connection proposition because x and z are free. A multiple empirical or mathematical connection proposition is a bound one if all term variables in it are bound. For example, lock(x)$\leqslant^{-1}$key(y)/$\wedge^{-1}$unlock(y, x) is a bound first-order multiple empirical or mathematical connection proposition because all term variables

in it are bound. In the simplest multiple empirical or mathematical connection propositions that we study, all term variables are relevantly bound.

Establishment by inductive composition of a multiple empirical or mathematical connection proposition can proceed only if the proposition is meaningful and bound.

5.8 Mutually inverse multiple diagrams for multiple empirical or mathematical connection propositions

There exist two mutually inverse problems between a multiple empirical or mathematical connection proposition and a mutually inverse multiple diagram: given a toue first-order multiple empirical or mathematical connection proposition we can draw one of its success diagram; and given a mutually inverse multiple diagram, we can determine whether a meaningful and bound first-order multiple empirical or mathematical connection proposition is true or not, this is establishment by explicit inductive composition of the proposition.

The three steps listed below describe the procedure to draw the mutually inverse multiple diagram from a true first-order multiple empirical or mathematical connection proposition:

(1) draw the term space corresponding to the proposition;

(2) investigate only the conjunction set of the sets corresponding to all the property fact propositions;

(3) in the conjunction set, if the non-property fact proposition is a positive one, then mark the zeroth-level elements in its corresponding set with "○", and mark the zeroth-level elements in its negation set with "×"; if the non-property fact proposition is a negative one, then mark the zeroth-level elements in its corresponding set with"×", and mark the zeroth-level elements in its negation set with "○".

Example 5.4: Draw the mutually inverse multiple diagram of the true first-order multiple empirical or mathematical connection proposition natural_number(x)/$\wedge^{-1}$ {natural_number(y)$\leqslant^{-1}x\leqslant y$}.

Solution:

Step (1) Draw the term space shown in Fig. 5.25.

Step (2) Consider only the conjunction set of ⟨natural_number(x), T⟩ and ⟨natural_number(y), T⟩

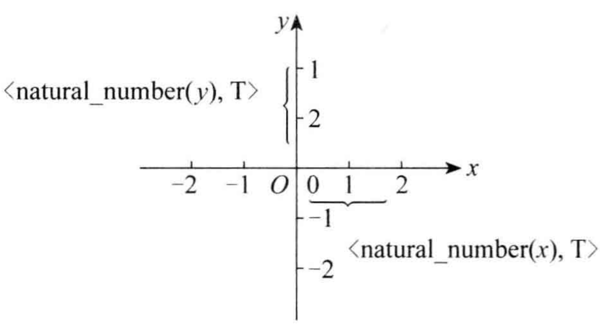


Fig. 5.25 Term space

corresponding to the property fact propositions natural_number(x) and natural_number(y), respectively, shown in Fig. 5.26.

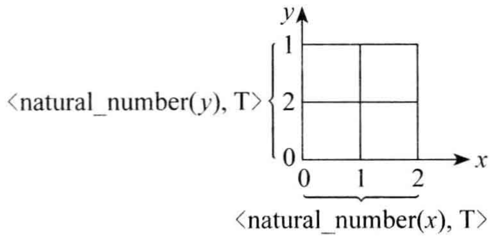


Fig. 5.26 Conjunction set

Step (3) Mark the zeroth-level elements in the set ⟨$x \leqslant y$, T⟩ with "○", and ⟨$x \leqslant y$, F⟩ with "×", as shown in Fig. 5.27.

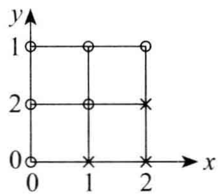


Fig. 5.27 Natural_number(x)/$\wedge^{-1}${natural_number(y)$\leqslant^{-1}$$x \leqslant y$}

Example 5.5: Given the mutually inverse multiple diagram shown in Fig. 5.28, determine whether it is a success or failure diagram of the following meaningful and bound first-order multiple empirical or mathematical connection propositions:

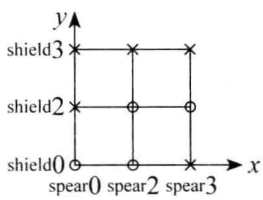


Fig. 5.28 A mutually inverse multiple diagram

(1) spear(x)$\leqslant^{-1}$shield(y)/$\wedge^{-1}$pierce(x, y) (every spear can pierce some shields);

(2) shield(y)/$\wedge^{-1}${spear(x)$\leqslant^{-1}$$\neg$pierce($x$, y) (there exist some shields such that no spear can pierce them);

(3) spear(x)/$\wedge^{-1}${shield(y)$\leqslant^{-1}$$\neg$pierce($x$, y) (there exist some spears such that they can pierce no shield);

(4) shield(y)$\leqslant^{-1}$spear(x)/$\wedge^{-1}$pierce(x, y) (for every shield there exist some spears that can pierce it).

Solution: Fig. 5.28 is a success diagram of (1), a success diagram of (2), a failure diagram of (3), a failure diagram of (4).

5.9 Proposition chains, property proposition segment chains, and least success diagrams of multiple empirical or mathematical connection propositions

There are four proposition chains of second-order multiple empirical or mathematical connection propositions with two property fact propositions shown in Figs. 5.29 to 5.32. For each proposition in the chains, a least success diagram is given at the right of it. The proposition chains of Figs. 5.29 and 5.30 can be merged into the property proposition segment chain shown in Fig. 5.33, and that of Figs. 5.31 and 5.32 into the one shown in Fig. 5.34.

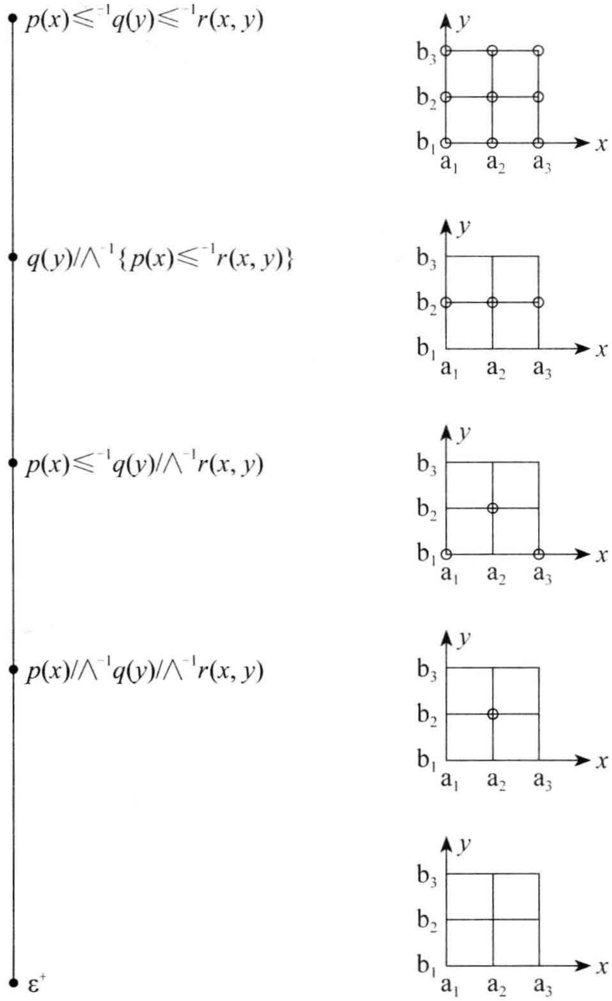


Fig. 5.29 Proposition chain of second-order multiple empirical or mathematical connection propositions with positive non-property fact propositions

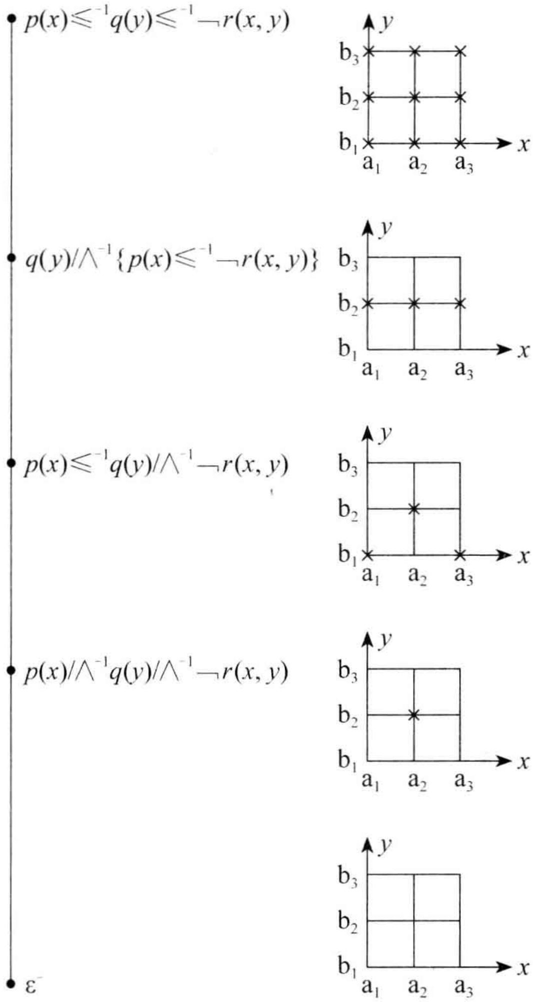


Fig. 5.30 Proposition chain of second-order multiple empirical or mathematical connection propositions with negative non-property fact propositions

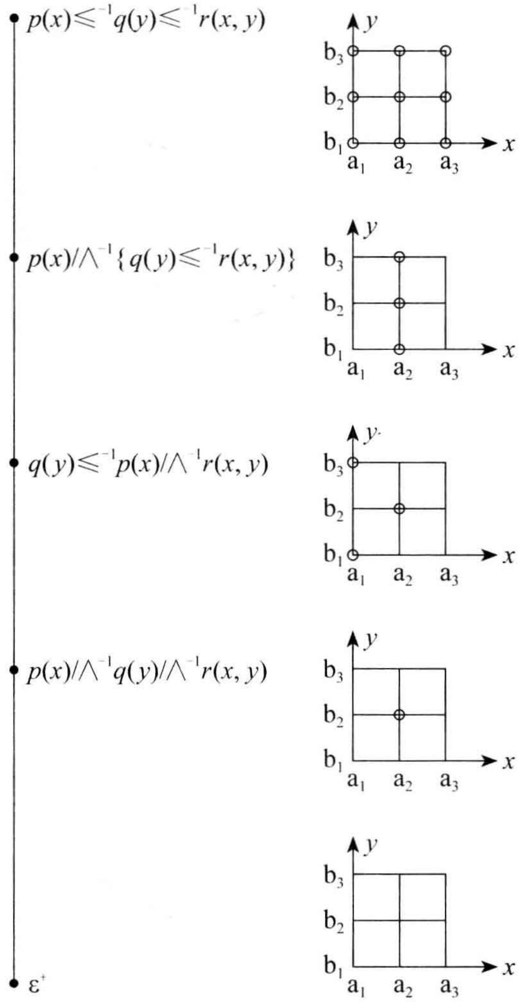


Fig. 5.31 Proposition chain of second-order multiple empirical or mathematical connection propositions with positive non-property fact propositions

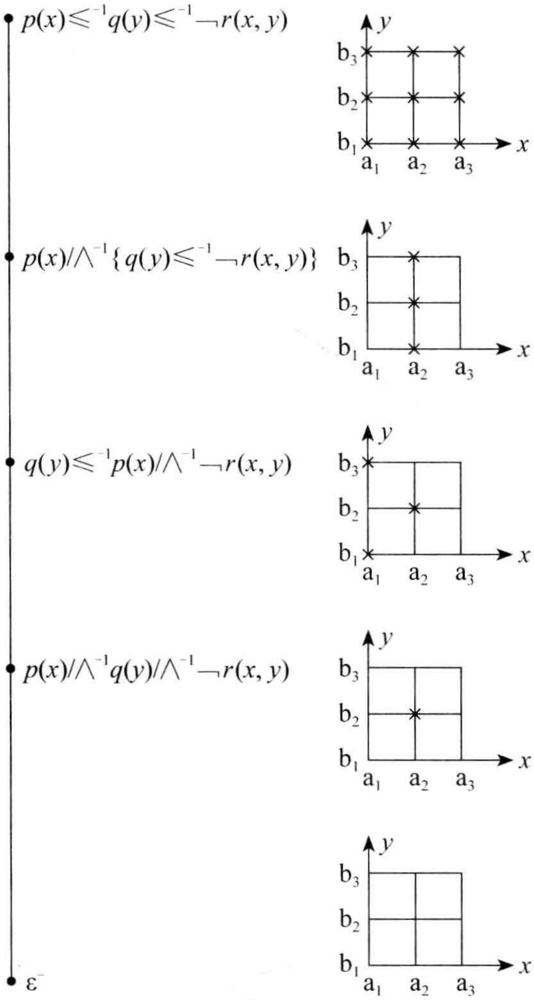


Fig. 5.32 Proposition chain of second-order multiple empirical or mathematical connection propositions with negative non-property fact propositions

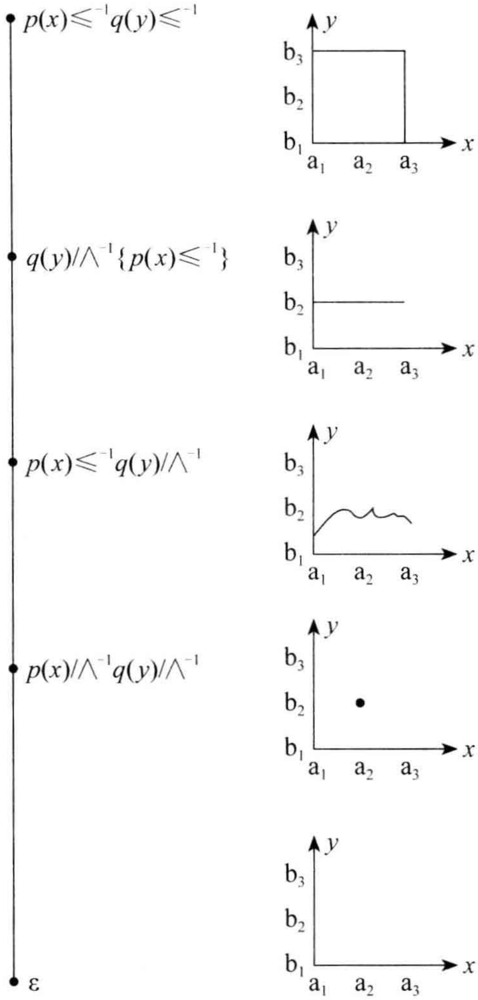


Fig. 5.33 Property proposition segment chain of second-order empirical or mathematical connection propositions

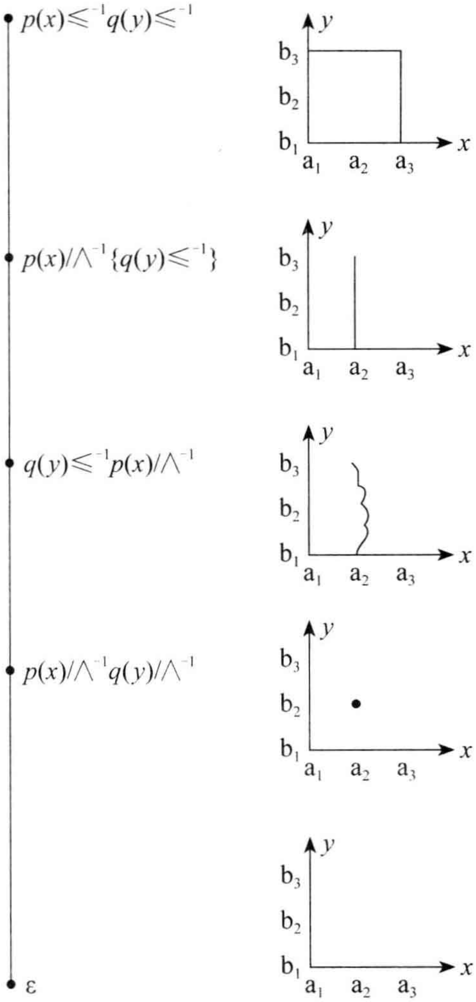


Fig. 5.34 Property proposition segment chain of second-order empirical or mathematical connection propositions

2.10 Decomposition system of first-level multiple predicate calculus

The grounds and inference rules of the decomposition system of the first-level multiple predicate calculus is the same as those of the decomposition system of the first-level single quasi-predicate calculus.

Example 5.6: The known: 3 is a natural number; for every natural number there exists one no greater than it. The unknown: there exists a natural number no greater than 3.

Solution:

The known:

natural_number(3),

natural_number(x)$\leqslant^{-1}$natural_number(y)/$\wedge^{-1}$ $y\leqslant x$.

The unknown:

natural_number(y)/$\wedge^{-1}$$y\leqslant 3$.

Proof:

(1) natural_number(3) P

(2) natural_number(x)$\leqslant^{-1}$natural_number(y)/ $\wedge^{-1}$$y\leqslant x$ P

(3) natural_number(y)/$\wedge^{-1}$$y\leqslant 3$ T(1)(2) Table 1.3

Chapter 6 Second-level multiple predicate calculus

Second-level multiple predicate calculus studies the relationship between multiple empirical or mathematical connection propositions and multiple logical connection propositions.

We regard multiple logical connection propositions as the generalization of single logical connection propositions. Therefore, we first study the features of single logical connection propositions, then we generalize them to multiple logical connection propositions.

Among single empirical or mathematical connection propositions, we have the square of opposition shown in Fig. 6.1.

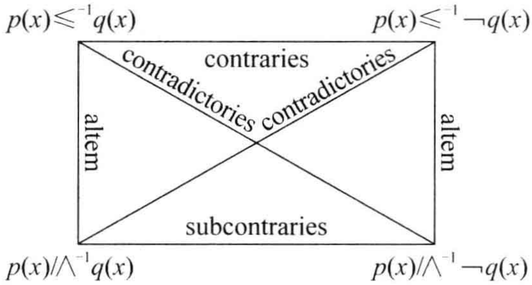


Fig. 6.1 Square of opposition

6.1 Meaningful and bound multiple logical connection propositions

A multiple logical connection proposition is a meaningless one if at least one multiple empirical or mathematical connection proposition in it is meaningless. For example, $\{p(x)\leqslant^{-1}\neg p(x)/\wedge^{-1}r(x, y)\}\leqslant^{-1}\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}r(x, y)\}\}$ is a meaningless second-order multiple logical connection proposition because both $p(x)$ and $\neg p(x)$ occur in the antecedent. If both multiple empirical or mathematical connection propositions in a multiple logical connection proposition are meaningful ones, then the multiple logical connection proposition is a meaningful one; e.g., $\{p(x)\leqslant^{-1}q(y)\leqslant^{-1}r(x, y)\}\leqslant^{-1}\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}$ is a meaningful second-order multiple logical connection proposition.

A multiple logical connection proposition is a free one if at least one fact proposition in it is free. For example, $\{p(x)\leqslant^{-1}q(y)\leqslant^{-1}r(x, y)\}\leqslant^{-1}\{s(y)/\wedge^{-1}\{p(x)\leqslant^{-1}r(x, y)\}\}$ is a free second-order multiple logical connection proposition, because both $q(y)$ and $s(y)$ are free second-order fact propositions. A multiple logical connection proposition is a bound one if all fact propositions in it are bound. For example, $\{p(x)\leqslant^{-1}q(y)/\wedge^{-1}r(x, y)\}\leqslant^{-1}\{p(x)/\wedge^{-1}$

$\{q(y) \leqslant^{-1} r(x, y)\}\}$ is a bound proposition because all fact propositions in it are bound. In the simplest multiple logical connection proposition we study in this book, there are only relevantly bound fact propositions.

Establishment by inductive composition of a multiple logical connection proposition can proceed only if the proposition is meaningful and bound.

6.2 Establishment by implicit inductive composition of mutually inverse implication propositions

In Fig. 6.1 $p(x) \leqslant^{-1} q(x)$ is the upper proposition of $p(x)/\wedge^{-1} q(x)$. In the proposition chain shown in Fig. 5.21, $p(x) \leqslant^{-1} q(x)$ is also the upper proposition of $p(x)/\wedge^{-1} q(x)$. From Section 5.5 we learn that the upper proposition mutually inversely implies the lower proposition, and $\{p(x) \leqslant^{-1} q(x)\} \leqslant^{-1} \{p(x)/\wedge^{-1} q(x)\}$ is a single logical theorem.

Generalizing the idea of a mutually inverse implication proposition of single empirical or mathematical connection propositions into multiple logical connection propositions, we have: suppose A and B are two multiple empirical or mathematical connection propositions on the same proposition chain, and A is the upper proposition of B, then $A \leqslant^{-1} B$ is a multiple logical theorem.

Example 6.1: Prove the meaningful and bound second-order multiple logical connection proposition $\{p(x)/\wedge^{-1}\{q(y) \leqslant^{-1} r(x, y)\}\} \leqslant^{-1} \{q(y) \leqslant^{-1} p(x)/\wedge^{-1} r(x, y)\}$.

Proof: $P(x)/\wedge^{-1}\{q(y) \leqslant^{-1} r(x, y)\}$ and $q(y) \leqslant^{-1} p(x)/\wedge^{-1} r(x, y)$ are on the same proposition chain, shown in Fig. 5.31, and the former is the upper proposition of the latter. Therefore, the proposition is proved.

6.3 Establishment by implicit inductive composition of contradictory propositions

Of the two contradictory propositions shown in Fig. 6.1, exactly one is true, and one is false; they can be neither both true nor both false. For example, $p(x) \leqslant^{-1} q(x)$ and $p(x)/\wedge^{-1} \neg q(x)$ are contradictory propositions; they can be neither both true nor both false, and therefore, $\{p(x) \leqslant^{-1} q(x)\} \oplus^{-1} \{p(x)/\wedge^{-1} \neg q(x)\}$ or $\neg\{p(x) \leqslant^{-1} q(x)\} =^{-1} p(x)/\wedge^{-1} \neg q(x)$ is a single logical theorem.

The contradictory propositions $p(x) \leqslant^{-1} q(x)$ and $p(x)/\wedge^{-1} \neg q(x)$ have two features diagrammatically: First, each least success diagram of $p(x) \leqslant^{-1} q(x)$ intersects with each least success diagram of $p(x)/\wedge^{-1} \neg q(x)$ at a zeroth-level element of the set $\langle p(x), T\rangle$ shown in Fig. 6.2. In Fig. 6.2, they intersect at the zeroth-level element marked both "o" and "×".

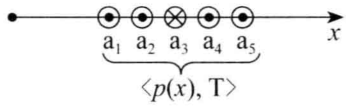


Fig. 6.2 Least success diagrams intersect each other

Secondly, if for one of $p(x)\leqslant^{-1}q(x)$ and $p(x)/\wedge^{-1}\neg q(x)$, we adopt a least success diagram of its adjacent lower proposition, and for the other, we still adopt its own least success diagram, then we can always find a couple of least success diagrams that do not intersect at a zeroth-level element of the set ⟨$p(x)$, T⟩.

Suppose, for $p(x)\leqslant^{-1}q(x)$, we adopt the least success diagram of its adjacent lower proposition shown in Fig. 5.21(2); and for $p(x)/\wedge^{-1}\neg q(x)$, we still adopt its own least success diagram; we then obtain Fig. 6.3 where the two least success diagrams do not intersect at a zeroth-level element of ⟨$p(x)$, T⟩.

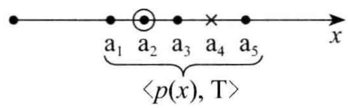


Fig. 6.3 Least success diagram of adjacent lower proposition does not intersect

Suppose, for $p(x)\leqslant^{-1}q(x)$, we adopt its own least success diagram; and for $p(x)/\wedge^{-1}$ $\neg q(x)$, we adopt the least success diagram of its adjacent lower proposition shown in Fig. 5.22(3); we then obtain Fig. 6.4 where the two least success diagrams do not intersect at a zeroth-level element of ⟨$p(x)$, T⟩.

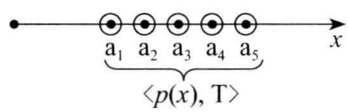


Fig. 6.4 Least success diagram of adjacent lower proposition does not intersect

$P(x)\leqslant^{-1}\neg q(x)$ and $p(x)/\wedge^{-1}q(x)$ are also contradictory propositions; they also have the above features.

Generalizing the idea of the contradictory single empirical or mathematical connection propositions into multiple logical connection propositions, we have: suppose A and B are multiple empirical or mathematical connection propositions having the same property fact propositions $p_1(x_1)$, …, $p_n(x_n)$ (this does not mean they have the same property fact proposition segment; e.g., $p_i(x_i)$ in A can be $p_i(x_i)\leqslant^{-1}$, while $p_i(x_i)$ in B can be $p_i(x_i)/\wedge^{-1}$, and rest can be inferred by analogy) which are neither permanently false nor permanently true and in which p_i is not $\neg p_j$ for any i and j, and the opposite non-property fact propositions which are neither permanently false nor permanently true; if, first, any least success diagram of A intersects

with any least success diagram of B at a zeroth-level element of the conjunction set of $p_1(x_1), \ldots, p_n(x_n)$; secondly, of A and B, one is adopted a least success diagram of its adjacent lower proposition, and the other is adopted a least success diagram of its own, and at least two least success diagrams can be found that do not intersect at a zeroth-level element of the conjunction set of $p_1(x_1),\ldots,p_n(x_n)$, then A and B are contradictory propositions and $A\oplus^{-1} B$, $\neg A=^{-1}B$ are multiple logical theorems.

Example 6.2: Prove the law of contradiction $\neg\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}=^{-1}q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$.

Proof: $\neg\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}=^{-1}q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$ is meaningful and bound, inductive composition can proceed.

First, on the conjunction set of $p(x)$ and $q(y)$, we draw a least success diagram of $q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)$, represented by the dotted line in Fig. 6.5, and one of $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$, represented by the continuing line in Fig. 6.5. The two least success diagrams intersect at the zeroth-level element (a_2, b_2) where $q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)$ requires $r(a_2, b_2)$ to be true, while $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$ requires $\neg r(a_2, b_2)$ to be true. It is verifiable that every least success diagram of $q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)$ intersects with every one of $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$ at a zeroth-level element of the conjunction set of $p(x)$ and $q(y)$.

Secondly, we adopt its own least success diagram for $q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)$, represented by the dotted line in Fig. 6.6 and adopt a least success diagram of its adjacent lower proposition for $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$, represented by the continuing line in Fig. 6.6. From Fig. 6.6 we observe that although the two least success diagrams intersect each other, the intersection point is not a zeroth-level element; i.e., the two least success diagrams do not intersect at a zeroth-level element of the conjunction set of $p(x)$ and $q(y)$. Hence, the law is proved.

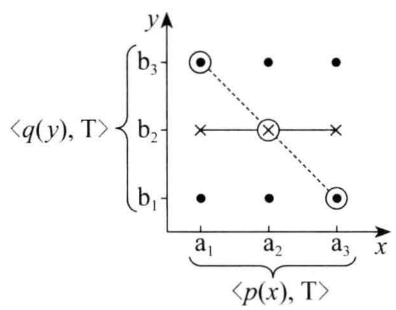


Fig. 6.5 Least success diagrams intersect each other

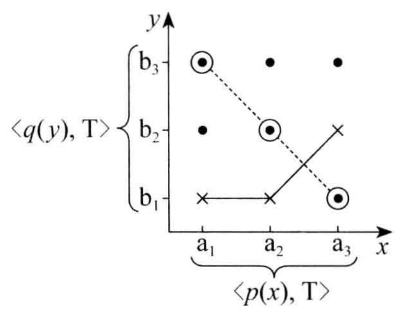


Fig. 6.6 Least success diagram of adjacent lower proposition does not intersect

6.4 Establishment by implicit inductive composition of contrary propositions

In Fig. 6.1, $p(x)\leqslant^{-1}q(x)$ and $p(x)\leqslant^{-1}\neg q(x)$ are contrary propositions that can be both false but not both true; therefore, $\neg\{p(x)\leqslant^{-1}q(x)\}\vee/^{-1}\neg\{p(x)\leqslant^{-1}\neg q(x)\}$ or $\neg\{\{p(x)\leqslant^{-1}q(x)\}/\wedge^{-1}\{p(x)\leqslant^{-1}\neg q(x)\}\}$ is a single logical theorem.

The contrary propositions $p(x)\leqslant^{-1}q(x)$ and $p(x)\leqslant^{-1}\neg q(x)$ have two features diagrammatically: first, each least success diagram of $p(x)\leqslant^{-1}q(x)$ and each one of $p(x)\leqslant^{-1}\neg q(x)$ intersect at a zeroth-level element of $p(x)$, shown in Fig. 6.7.

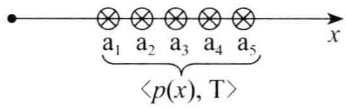


Fig. 6.7 Least success diagrams intersect each other

Secondly, if for one of $p(x)\leqslant^{-1}q(x)$ and $p(x)\leqslant^{-1}\neg q(x)$ we adopt a least success diagram of its adjacent lower proposition, and for the other we still adopt its own least success diagram, then any pair of the least success diagrams still intersect at a zeroth-level element of $p(x)$. Suppose, for $p(x)\leqslant^{-1}q(x)$ we adopt a least success diagram of its adjacent lower proposition, for $p(x)\leqslant^{-1}\neg q(x)$ we still adopt its own least success diagram, shown in Fig. 6.8, then the two least success diagrams still intersect each other at a zeroth-level element of $p(x)$.

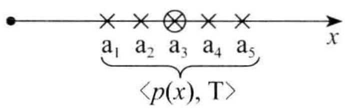


Fig. 6.8 Least success diagram of adjacent lower proposition still intersect

Generalizing the idea of the contrary single empirical or mathematical connection propositions into multiple logical connection propositions, we have: suppose A and B are multiple empirical or mathematical connection propositions having the same property fact propositions $p_1(x_1)$, …, $p_n(x_n)$ which are neither permanently false nor permanently true and in which p_i is not $\neg p_j$ for any i and j, and the opposite non-property fact propositions which are neither permanently false nor permanently true; if, first, any least success diagram of A intersects with any least success diagram of B at a zeroth-level element of the conjunction set of $p_1(x_1)$, …, $p_n(x_n)$; secondly, of A and B, one is adopted a least success diagram of its adjacent lower proposition, and the other is adopted a least success diagram of its own, and any such pair of least success diagrams still intersect each other at a zeroth-level element of the conjunction set of $p_1(x_1)$, …, $p_n(x_n)$, then A and B are contrary propositions and $\neg A\vee/^{-1}$

$\neg$B, $\neg\{A/\wedge^{-1}B\}$ are multiple logical theorems.

Example 6.3: Prove the law of contrariety $\neg\{\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}/\wedge^{-1}\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}\}\}$.

Proof: $\neg\{\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}/\wedge^{-1}\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}\}\}$ is meaningful and bound, inductive composition can proceed. First, on the conjunction set of $p(x)$ and $q(y)$, we draw a least success diagram of $p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}$, represented by the continuing line in Fig. 6.9, and one of $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$, represented by the dotted line in Fig. 6.9. The two least success diagrams intersect at the zeroth-level element (a_2, b_2). It is verifiable that every least success diagram of $p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}$ intersects with every one of $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$ at a zeroth-level element of the conjunction set of $p(x)$ and $q(y)$.

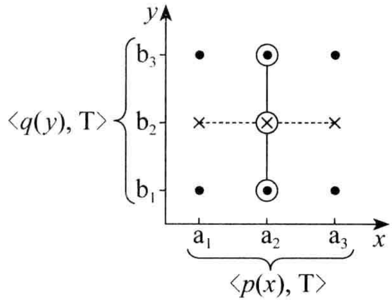


Fig. 6.9 Least success diagrams intersect each other

Secondly, we adopt a least success diagram of its adjacent lower proposition for $p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}$, represented by the continuing line in Fig. 6.10, and adopt its own least success diagram for $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}$, represented by the dotted line in Fig. 6.10. The two least success diagrams still intersect at the zeroth-level element (a_2, b_2). It is verifiable that every such pair of least success diagrams intersects each other. Hence, the law is proved.

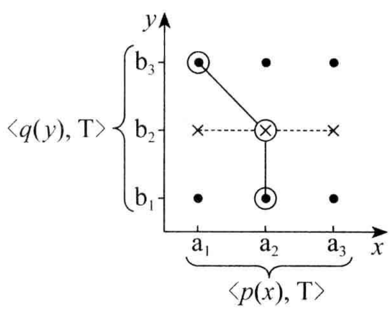


Fig. 6.10 Least success diagram of adjacent lower proposition still intersects

Around the third century B.C., in the warring states of ancient China, there was a merchant selling weapons. He boasted his spear by saying, "My spear is so sharp that it can pierce

through all shields."Later, he boasted about his shield by saying, "My shield is so strong that no spear can pierce through it." Then, a bystander asked, "What would happen if you use your spear to pierce your shield?" The merchant cannot answer the question. Later, the ancient Chinese philosopher Han Fei summarized a law called the law of contradiction from the story: "A spear that can pierce through all shields and a shield that no spear can pierce through cannot coexist." In mutually-inversistic logic, this law is denoted by: $\neg\{\{\text{spear}(x)/\wedge^{-1}\{\text{shield}(y)\leqslant^{-1}\text{pierce}(x, y)\}\}/\wedge^{-1}\{\text{shield}(y)/\wedge^{-1}\{\text{spear}(x)\leqslant^{-1}\neg\text{pierce}(x, y)\}\}\}$ which is an instance of the law of contrariety $\neg\{\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}/\wedge^{-1}\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}\neg r(x, y)\}\}\}$ when p is assigned a spear, q a shield, r pierce.

6.5 Establishment by implicit inductive composition of subcontrary propositions

In Fig. 6.1, $p(x)/\wedge^{-1}q(x)$ and $p(x)/\wedge^{-1}\neg q(x)$ are subcontrary propositions that can be both true but not both false; therefore, $\{p(x)/\wedge^{-1}q(x)\}\vee/^{-1}\{p(x)/\wedge^{-1}\neg q(x)\}$ is a single logical theorem.

The subcontrary propositions $p(x)/\wedge^{-1}q(x)$ and $p(x)/\wedge^{-1}\neg q(x)$ have the diagrammatic feature that at least a least success diagram of $p(x)/\wedge^{-1}q(x)$ and one of $p(x)/\wedge^{-1}\neg q(x)$ can be found that do not intersect at a zeroth-level element of $p(x)$, shown in Fig. 6.11.

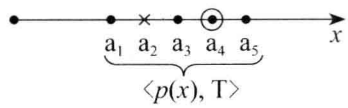


Fig. 6.11 Least success diagrams do not intersect each other

Generalizing the idea of the subcontrary single empirical or mathematical connection propositions into multiple logical connection propositions, we have: suppose A and B are multiple empirical or mathematical connection propositions having the same property fact propositions $p_1(x_1)$, …, $p_n(x_n)$ which are neither permanently false nor permanently true and in which p_i is not $\neg p_j$ for any i and j, and the opposite non-property fact propositions which are neither permanently false nor permanently true; if at least a pair of least success diagrams of A and B can be found that do not intersect at a zeroth-level element of the conjunction set of $p_1(x_1)$, …, $p_n(x_n)$, then A and B are subcontrary propositions and $A/^{-1}B$ is a multiple logical theorem.

Example 6.4: Prove the meaningful and bound second-order multiple logical connection proposition $\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}\vee/^{-1}\{p(x)\leqslant^{-1}q(y)/\wedge^{-1}\neg r(x, y)\}$.

Proof: Draw a least success diagram of $q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)$, represented by the continuing line in Fig. 6.12 and one of $p(x)\leqslant^{-1}q(y)/\wedge^{-1}\neg r(x, y)$, represented by the dotted line

in Fig. 6.12 on the conjunction set of $p(x)$ and $q(y)$. From Fig. 6.12 we observe that the two least success diagrams do not intersect at a zeroth-level element of the conjunction set of $p(x)$ and $q(y)$. Hence, the proposition is proved.

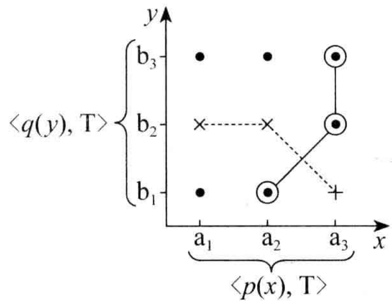


Fig. 6.12 Least success diagrams do not intersect each other

6.6 Square of opposition among multiple empirical or mathematical connection propositions

Similar to Fig. 6.1, we have the square of opposition among multiple empirical or mathematical connection propositions shown in Fig. 6.13.

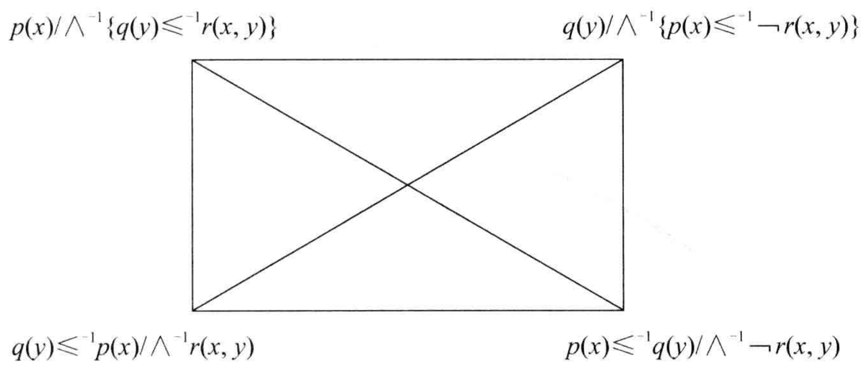


Fig. 6.13 Logical square for multiple empirical or mathematical connection propositions

6.7 Decomposition system of second-level multiple predicate calculus

The grounds and inference rules of the decomposition system of the second-level multiple predicate calculus is the same as those of the decomposition system of the first-level single quasi-predicate calculus.

Example 6.5: The known: all natural numbers are comparable. The unknown: some natural numbers are comparable.

Solution:

Suppose: n_n(x): x is a natural number.

comparable(x, y): x and y are comparable.

The known:

n_n(x)$\leqslant^{-1}$n_n(y) $\leqslant^{-1}$comparable(x, y),

$\{p(x)\leqslant^{-1}q(y)\leqslant^{-1}r(x, y)\}\leqslant^{-1}\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}$,

$\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}\leqslant^{-1}\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}$,

$\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}\leqslant^{-1}\{q(y)/\wedge^{-1}p(x)/\wedge^{-1}r(x, y)\}$.

The unknown:

n_n(y)/$\wedge^{-1}$n_n(x)/$\wedge^{-1}$comparable(x, y).

Proof:

(1) n_n(x)$\leqslant^{-1}$n_n(y)$\leqslant^{-1}$comparable(x, y)	P
(2) $\{p(x)\leqslant^{-1}q(y)\leqslant^{-1}r(x, y)\}\leqslant^{-1}\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}$	P
(3) n_n(x)/$\wedge^{-1}${n_n(y)$\leqslant^{-1}$comparable(x, y)}	T (1) (2) Table 1.3
(4) $\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}r(x, y)\}\}\leqslant^{-1}\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}$	P
(5) n_n(y)$\leqslant^{-1}$n_n(x)/$\wedge^{-1}$comparable(x, y)	T (3) (4) Table 1.3
(6) $\{q(y)\leqslant^{-1}p(x)/\wedge^{-1}r(x, y)\}\leqslant^{-1}\{q(y)/\wedge^{-1}p(x)/\wedge^{-1}r(x, y)\}$	P
(7) n_n(y)/$\wedge^{-1}$n_n(x)/$\wedge^{-1}$comparable(x, y)	T (5) (6) Table 1.3

Chapter 7 Mutually-inversistic propositional calculus

7.1 Formation of fact propositions

An assertion is a declarative sentence. A zeroth-order fact proposition is an assertion. "On Aug. 20, 2006, in Xi'an, it rained" is a zeroth-order fact proposition, denoted by U_1^1. U_1^1 is true, denoted by $\langle U_1^1, T\rangle$.

U_1^2: On Oct. 1, in Beijing, it rained.

U_1^2 is false, denoted by $\langle U_1^2, F\rangle$. Zeroth-order fact propositions are invariable propositions, denoted by U, V, and W with subscript and superscript.

U_1: it rains.

U_1 is a first-order fact proposition, which is a variable proposition, ranging over zeroth-order fact propositions. For example, U_1 ranges over U_1^1 and U_1^2. First-order fact propositions are denoted by U, V, and W with subscript.

U, *V*, and *W* are second-order fact propositions, which are variable propositions, ranging over first-order fact propositions. For example, *U* ranges over U_1, U_2 (the wind blows), U_3 (the ground is wet), U_4 (the road is slippery).

Single empirical or mathematical connection propositions, quasi-logical connection propositions, single logical connection propositions, and quasi-transcendent logical connection propositions in mutually-inversistic propositional calculus are similar to those in mutually-inversistic predicate calculus.

7.2 First-level single quasi-propositional calculus

First-level single quasi-propositional calculus studies the relationship between fact propositions and single empirical or mathematical connection propositions, quasi-logical connection propositions.

Example 7.1: Truthify $U_1 \leqslant^{-1} U_3$ (if it rains, then the ground is wet) by explicit inductive composition.

Solution: U_1 is not permanently false, for on Aug. 20, in Xi'an, it rained (U_1^1). U_3 is not permanently true, for on Oct. 1, in Beijing, the ground is not wet (U_3^2). Therefore, $U_1 \leqslant^{-1} U_3$ is meaningful, inductive composition can go on.

The operation table of inductive composition for $U_1 \leqslant^{-1} U_3$ is given in Table 7.1.

Table 7.1 Operation table of inductive composition for $U_1 \leqslant^{-1} U_3$

$\langle U_1, F\rangle\langle U_3, F\rangle$	$\langle U_1 \leqslant^{-1} U_3, T\rangle$
$\langle U_1, F\rangle\langle U_3, T\rangle$	$\langle U_1 \leqslant^{-1} U_3, n\rangle$
$\langle U_1, T\rangle\langle U_3, F\rangle$	$\langle U_1 \leqslant^{-1} U_3, F\rangle$
$\langle U_1, T\rangle\langle U_3, T\rangle$	$\langle U_1 \leqslant^{-1} U_3, T\rangle$

In Table 7.1, the four arguments are no longer minsets, but minareas, because they are composed of instances, not elements. For example, $\langle U_1^1, T\rangle\langle U_3^1, T\rangle$ is an instance, not an element of $\langle U_1, T\rangle\langle U_3, T\rangle$. The first and fourth rows of Table 7.1 constitute the support area of $U_1 \leqslant^{-1} U_3$, the third row constitutes the opposition area, and the second row constitutes the neutral area. Inductive composition is performed as follows:

On Aug. 20, 2006, in Xi'an, it rained, and the ground was wet. This is the fourth row of Table 7.1. On Oct. 1, 2006, in Beijing, it did not rain, and the ground was not wet. This is the first row of Table 7.1. It is not the case that it rains and the ground is not wet. That is, the third row of Table 7.1 never occurs.

Note that the instances of the antecedent and the consequent should be in the same time and place, under the same condition. For example, when the instance of the antecedent is "on Aug. 20, 2006, in Xi'an, it rained", the instance of the consequent should be "on Aug. 20, 2006, in Xi'an, the ground was wet". It will not do when the instance of the antecedent is "on Aug. 20, 2006, in Xi'an, it rained", while the instance of the consequent is "on Oct. 1, 2006, in Beijing, the ground was not wet". This requirement is the boundness requirement.

After arbitrarily investigating a finite number of instances, the inductive composition concludes. At this time, in order to determine whether the unknown single empirical or mathematical connection proposition is true or not, we need to investigate the following two criteria:

(1) The minareas constitute the opposition area are all empty;

(2) None of the minareas constitute the support area is empty.

If both criteria are met, then the proposition under investigation is true; otherwise, it is false. In this example, both criteria are met, therefore, $U_1 \leqslant^{-1} U_3$ is truthified.

After being truthified, "if it rains, then the ground is wet" serve as the major premise. "On Aug. 20, 2006, in Xi'an, it rained" serve as the minor premise. Using Table 1.3, we can infer the conclusion "on Aug. 20, 2006, in Xi'an, the ground was wet."

The inductive composition of quasi-logical connection proposition in propositional calculus is similar to that in predicate calculus.

The antecedent "it rains" and the consequent "the ground is wet" of the single empirical

or mathematical theorem "if it rains, then the ground is wet" have only the nexus of semantics, not the nexus of syntactic. After we have thrthified "if it rains, then the ground is wet" and have determined "on Aug. 20, 2006, in Xi'an, it rained"; before we use Table 1.3 to infer "on Aug. 20, 2006, in Xi'an, the ground was wet" and make an on-the-spot investigation, we have not determined "on Aug. 20, 2006, in Xi'an, the ground was wet". This means that the minor premise alone cannot decide the conclusion. This property is called the first-level non-decidedness of the minor premise to the conclusion. If this property holds, then the conclusion is new knowledge to the minor premise.

The antecedent "it rains" and the consequent "it rains or cows eat grass" of the instance "if it rains, then it rains or cows eat grass" of the quasi-logical theorem $U\leqslant^{-1}U\vee V$ have not only the nexus of semantics but also the nexus of syntactic: both the antecedent and the consequent have the fact proposition "it rains". After we have truthified "if it rains, then it rains or cows eat grass" and have determined "on Aug. 20, 2006, in Xi'an, it rained"; before we use Table 1.3 to infer "on Aug. 20, 2006, in Xi'an, it rained or cows ate grass" and make on-the-spot investigation, we have already determined "on Aug. 20, 2006, in Xi'an, it rained or cows ate grass" to be true. This means that the minor premise alone can decide the conclusion, and the property of first-level non-decidedness of the minor premise to the conclusion does not hold, and the conclusion is not new knowledge to the minor premise.

The grounds and inference rules of the decomposition system of first-level single quasi-propositional calculus is the same as those of the decomposition system of first-level single quasi-predicate calculus.

Example 7.2: the known: on Aug. 20, 2006, in Xi'an, it rained (U_1^1); if it rains, then the ground is wet ($U_1\leqslant^{-1}U_3$); if the ground is wet, then the road is slippery ($U_3\leqslant^{-1}U_4$). The unknown: on Aug. 20, 2006, in Xi'an, the road was slippery (U_4^1).

Proof:

(1) U_1^1	P
(2) $U_1\leqslant^{-1}U_3$	P
(3) U_3^1	T (1) (2) Table1.3
(4) $U_3\leqslant^{-1}U_4$	P
(5) U_4^1	T (3) (4) Table1.3

The second-level single quasi-propositional calculus is similar to the second-level single quasi-predicate calculus.

Part 2
Mutually-inversistic set theory

Mutually-inversistic set theory is basically isomorphic with mutually-inversistic predicate calculus. Their distinctions lie in that mutually-inversistic predicate calculus studies the objective world from the intension and extension perspective, while mutually-inversistic set theory studies it from the extension perspective only; mutually-inversistic predicate calculus studies the mutual inverseness of inductive composition and decomposition, while mutually-inversistic set theory studies the main, the auxiliary (which are the foundations of mutually-inversistic abstract algebra), and mutually inverse coordinate systems hierarchy (which is the foundation of mutually-inversistic mathematics).

Chapter 8
Fundamentals of mutually-inversistic set theory

8.1 Set operators

The 12 set operators corresponding to logical operators are: $\sim$((absolute) complement), $\cap$(intersection), $\cup$((compatible) union), $\oplus$(incompatible union, Boolean sum), $|\cap^{-1}$ (mutually inverse intersection), $\cup|^{-1}$ (mutually inverse union), $\subseteq^{-1}$ (being mutually inversely contained, subset), $=^{-1}$ (mutually inverse equivalence), $\subset^{-1}$ (being mutually inversely properly contained, proper subset), $\cup^{-1}$ (mutually inverse compatible union), $\oplus^{-1}$ (mutually inverse incompatible union), $\times^{-1}$ (mutually inverse intercross). In mutually-inversistic set theory, two more composition operators are introduced: $\otimes$(Boolean product), and — (relative complement). Their operation tables are shown in Tables 8.1 and 8.2.

Table 8.1 Operation table for $\otimes$

$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \in \mathrm{A} \otimes \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} \otimes \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} \otimes \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \in \mathrm{A} \otimes \mathrm{B}$

Table 8.2 Operation table for —

$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} — \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} — \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \in \mathrm{A} — \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} — \mathrm{B}$

The operation tables for $\subseteq^{-1}$ are shown in Tables 8.3 to 8.5.

Table 8.3 Operation table of inductive composition for $\subseteq^{-1}$

$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \in \mathrm{A} \subseteq^{-1} \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \notin \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	n
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \notin \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \notin \mathrm{A} \subseteq^{-1} \mathrm{B}$
$(\sigma_1, \ldots, \sigma_j) \in \mathrm{A}$	$(\sigma_i, \ldots, \sigma_k) \in \mathrm{B}$	$(\sigma_1, \ldots, \sigma_j, \sigma_i, \ldots, \sigma_k) \in \mathrm{A} \subseteq^{-1} \mathrm{B}$

Table 8.4 Operation table of decomposition one for $\subseteq^{-1}$

$\langle A\subseteq^{-1}B, F\rangle$	$(\sigma_l, \ldots, \sigma_j)\notin A$	u
$\langle A\subseteq^{-1}B, F\rangle$	$(\sigma_l, \ldots, \sigma_j)\in A$	u
$\langle A\subseteq^{-1}B, T\rangle$	$(\sigma_l, \ldots, \sigma_j)\notin A$	u
$\langle A\subseteq^{-1}B, T\rangle$	$(\sigma_l, \ldots, \sigma_j)\in A$	$(\sigma_i, \ldots, \sigma_k)\in B$

Table 8.5 Operation table of decomposition two for $\subseteq^{-1}$

$\langle A\subseteq^{-1}B, F\rangle$	$(\sigma_i, \ldots, \sigma_k)\notin B$	u
$\langle A\subseteq^{-1}B, F\rangle$	$(\sigma_i, \ldots, \sigma_k)\in B$	u
$\langle A\subseteq^{-1}B, T\rangle$	$(\sigma_i, \ldots, \sigma_k)\notin B$	$(\sigma_l, \ldots, \sigma_j)\notin A$
$\langle A\subseteq^{-1}B, T\rangle$	$(\sigma_i, \ldots, \sigma_k)\in B$	u

A set is composed of elements. Suppose element e_1 is a member of the set S, then we say "e_1 belongs to S", denoted by $e_1\in S$. Suppose e_2 is not a member of the set S, then we say "e_2 doesn't belong to S", denoted by $e_2\notin S$. One way of describing a set is by listing its elements between braces. Suppose set S is composed of e_1 and e_2, then we write $S=\{e_1, e_2\}$. The other way of describing a set is by describing the property its elements satisfy, e.g. $S=\{x|x \text{ is a natural number}\}$.

$\sim$A is a set whose elements are those that are in the universe U, but are not in A. For example, suppose U={a, b, c}, A={a}, then $\sim$A={b, c}. A∩B is a set whose elements are those that are in both A and B. For example, suppose A={a, b}, B={b, c}, then A∩B={b}. A∪B is a set whose elements are those that are in either A or B. For example, suppose A={a, b}, B={b, c}, then A∪B={a, b, c}. The difference between A∩B and $A|\cap^{-1}B$ is that A∩B is the intersection set, while $A|\cap^{-1}B$ is the intersection relationship or connection, meaning that A and B intersect each other, or that some A are B. Of course, there is certain relationship between A∩B and $A|\cap^{-1}B$: if A∩B is not empty, then $A|\cap^{-1}B$ holds; otherwise, $A|\cap^{-1}B$ does not hold. The difference between A⊆B in naïve set theory and $A\subseteq^{-1}B$ is that A⊆B only has establishment process, while $A\subseteq^{-1}B$ has establishment and employment processes. For example, suppose A={a, b}, B={a, b, c}, then A⊆B and $A\subseteq^{-1}B$ hold. This is the establishment process. A⊆B only has this process. While $A\subseteq^{-1}B$, in addition to this process, also has the employment process: from $A\subseteq^{-1}B$ holding and a∈A, we can infer a∈B.

8.2 Elements, sets, and propositions

Changing the logical operators in Section 1.2 into set operators, we obtain elements, sets, and propositions in mutually-inversistic set theory. Changing the logical operators in quasi-logical, single logical, multiple logical, quasi-transcendent logical connection propositions

into set operators, we obtain quasi-set connection propositions, single set connection propositions, multiple set connection propositions, quasi-transcendent set connection propositions. What follows is the description of the main thread: term—fact proposition—single empirical or mathematical connection proposition—single set connection proposition.

8.2.1 Zeroth-level elements

An n-tuple composed of n (n=1, 2, 3, ...) zeroth-order terms is a zeroth-level element constant; e.g., (1+1, 2) is a zeroth-level element constant. An n-tuple composed of n first-order terms is a zeroth-level element variable; e.g., (x, y) is a zeroth-level element variable.

8.2.2 Zeroth-level sets and first-level elements

A predicate acting on a zeroth-level element constant obtains a zeroth-order fact proposition, which is a belonging relationship of zeroth-level element constant to zeroth-level set constant. For example, = acting on (1+1, 2) obtains 1+1=2, which can be regarded as $(1+1, 2)\in x=y$.

A predicate acting on a zeroth-level element variable obtains a first-order fact proposition, which is a zeroth-level set constant. For example, =acting on (x, y) obtains $x=y$. It is a zeroth-level set constant, (1+1, 2) belongs to it, (3, 2) does not belong to it.

A second-order fact proposition is a zeroth-level set variable ranging over zeroth-level set constants. For example, P is a zeroth-level set variable ranging over $x=y$, man(x).

Two zeroth-level set constants constitute a first-level element constant; e.g., (int(x), rat(x)) is a first-level element constant. Two zeroth-level set variables constitute a first-level element variable; e.g., (P, Q) is a first-level element variable.

8.2.3 First-level sets and second-level elements

An empirical or mathematical connection operator acting on a first-level element constant obtains a first-order single empirical or mathematical connection proposition. For example, $\subseteq^{-1}$ acting on (int(x), rat(x)) obtains int(x)$\subseteq^{-1}$rat(x). A first-order single empirical or mathematical connection proposition is a belonging relationship of first-level element constant to first-level set constant. For example, int(x)$\subseteq^{-1}$rat(x) can be regarded as (int(x), rat(x))$\in P\subseteq^{-1}Q$. A first-order single empirical or mathematical connection proposition is also a connection between two zeroth-level set constants; e.g., int(x)$\subseteq^{-1}$rat(x) is a being mutually inversely contained connection between int(x) and rat(x).

An empirical or mathematical connection operator acting on a first-level element variable obtains a second-order single empirical or mathematical connection proposition. For example, $\subseteq^{-1}$ acting on $(P\cap Q, R)$ obtains $P\cap Q\subseteq^{-1}R$. An second-order single empirical or mathematical connection proposition is a first-level set constant. For example, $P\cap Q\subseteq^{-1}R$ is

a first-level set constant, (parent(x, y)$\cap$ancestor(y, z), ancestor(x, z)) belongs to it, ($x=y\cap y=z$, $x<z$) does not belong to it. A second-order single empirical or mathematical connection proposition is also a connection between two zeroth-level set variables. For example, $P\cap Q \subseteq^{-1} R$ is a being mutually inversely contained connection between $P\cap Q$ and R.

A third-order single empirical or mathematical connection proposition is a first-level set variable ranging over first-level set constants. For example, Ψ is a first-level set variable ranging over $P\subseteq^{-1}Q$, $P\cap Q\subseteq^{-1}R$.

Two first-level set constants constitute a second-level element constant; e.g., ($P\subseteq^{-1}Q$, $\sim Q\subseteq^{-1}\sim P$) is a second-level element constant. Two first-level set variables constitute a second-level element variable; e.g., (Ψ, Ω) is a second-level element variable.

8.2.4 Second-level sets

A set connection operator (corresponding to a logical connection operator) acting on a second-level element constant obtains a second-order single set connection proposition. For example, $=^{-1}$ acting on ($P\subseteq^{-1}Q$, $\sim Q\subseteq^{-1}\sim P$) obtains $\{P\subseteq^{-1}Q\}=^{-1}\{\sim Q\subseteq^{-1}\sim P\}$. A second-order single set connection proposition is a belonging relationship of second-level element constant to second-level set constant. For example, $\{P\subseteq^{-1}Q\}=^{-1}\{\sim Q\subseteq^{-1}\sim P\}$ can be regarded as ($P\subseteq^{-1}Q$, $\sim Q\subseteq^{-1}\sim P$)$\in\Psi=^{-1}\Omega$. A second-order single set connection proposition is also a connection between two first-level set constants; e.g., $\{P\subseteq^{-1}Q\}=^{-1}\{\sim Q\subseteq^{-1}\sim P\}$ is a mutually inverse equivalent connection between $P\subseteq^{-1}Q$ and $\sim Q\subseteq^{-1}\sim P$. A second-order single set connection proposition is also a connection of connection between two zeroth-level set variables. Take $\{P\subseteq^{-1}Q\}=^{-1}\{\sim Q\subseteq^{-1}\sim P\}$ as an example, P and Q are two zeroth-level set variables, $P\subseteq^{-1}Q$ is a connection between them, $\sim Q\subseteq^{-1}\sim P$ is another connection between them, $\{P\subseteq^{-1}Q\}=^{-1}\{\sim Q\subseteq^{-1}\sim P\}$ is a connection of the two connections.

A set connection operator acting on a second-level element variable obtains a third-order single set connection proposition. For example, $\subseteq^{-1}$ acting on ($\Psi\cap\Omega$, Θ) obtains $\Psi\cap\Omega\subseteq^{-1}\Theta$. A third-order single set connection proposition is a second-level set constant. For example, $\Psi\cap\Omega\subseteq^{-1}\Theta$ is a second-level set constant, ($\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}$, $\{P|\cap^{-1}R\}$) belongs to it, ($\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}$,$\{P|\cap^{-1}R\}$) does not belong to it. A third-order single set connection proposition is also a connection between two first-level set variables. For example, $\Psi\cap\Omega\subseteq^{-1}\Theta$ is a being mutually inversely contained connection between $\Psi\cap\Omega$ and Θ.

A fourth-order single set connection proposition is a second-level set variable ranging over second-level set constants. For example, Λ is a second-level set variable ranging over $\Psi=^{-1}\Omega$, $\Psi\cap\Omega\subseteq^{-1}\Theta$.

8.2.5 A three-storeyed building with foundation of elements and sets

Elements and sets are like a three-storeyed building with foundation, shown in Fig. 8.1.

0order	1order	2order	3order	4order	
		2BelRel $\{P=^{-1}Q\}\subseteq^{-1}\{P\subseteq^{-1}Q\}$	2SetConst $\Psi\subseteq^{-1}\Omega$	2SetVar Λ	2Set
		2EleConst $(P=^{-1}Q, P\subseteq^{-1}Q)$	2EleVar (Ψ, Ω)		2Ele
	1BelRel $\mathrm{int}(x)\subseteq^{-1}\mathrm{rat}(x)$	1SetConst $P\subseteq^{-1}Q$	1SetVar Ψ		1Set
	1EleConst $(\mathrm{int}(x), \mathrm{rat}(x))$	1EleVar (P, Q)			1Ele
0BelRel int(2)	0SetConst $\mathrm{int}(x)$	0SetVar P			0Set
0EleConst (2, 3)	0EleVar (x, y)				0Ele
TermConst 2	TermVar x				Term

Fig. 8.1 A three-storeyed building with foundation of elements and sets

In Fig. 8.1, 0BelRel stands for zeroth-level belonging relationship, 1EleConst stands for first-level element constant, 2SetVar stands for second-level set variable, rest can be inferred by analogy. In Fig. 8.1, terms are the foundation. Zeroth-level elements are the floors of the ground floor. Zeroth-level sets are the ceilings of the ground floor. Rest can be inferred by analogy. The vertical dotted lines partition the floors into rooms of different orders.

8.3 Empirical abstraction, mathematical abstraction, and set-theoretic abstraction

Human beings initially abstract empirically from nature elements John, Building No. 1, which are the object of study of empirical sciences. Later, they abstract mathematically from them elements 1, 2, 3, …, which are the object of study of mathematics. Finally, they abstract set-theoretically from them elements a, b, c, …, which are the object of study of set theory. From empirical abstraction through mathematical abstraction to set-theoretic abstraction, the extent of abstraction goes higher and higher.

Bob, Max, Sam constitute the set man(x), which is the object of study of empirical

sciences. 0, 1, 2 constitute the set integer(x), which is the object of study of mathematics. A, b, c constitute the set {a, b, c}, which is the object of study of set theory.

8.4 Mutually inverse coordinate systems hierarchy

Elements, sets and propositions can be described by mutually inverse coordinate systems hierarchy, which is composed of four coordinate systems: the coordinate system of terms, the coordinate system of facts, the coordinate system of single empirical or mathematical connections, the coordinate system of single set connections. Fig. 8.2 is a coordinate system of terms with empirical abstraction. Fig. 8.3 is a coordinate system of terms with mathematical abstraction, Fig. 8.4 to 8.6 are coordinate systems of facts with empirical, mathematical, set-theoretic abstractions respectively. Fig. 8.7 is a coordinate system of single empirical or mathematical connections. Fig. 8.8 is a coordinate system of single set connections.

8.4.1 Coordinate system of terms

Coordinate systems of terms are shown in Figs. 8.2 and 8.3. In these figures, O_t is the origin. X–, y– z–axes are the coordinate axes of terms. X, y, z are term variables. The points on x–, y–, z–axes are term constants. The space formed by the x–, y–, z–axes is the term space. A point in the term space is a 3-component zeroth-level element constant; e.g., point (Max, Bob, Sam) in Fig. 8.2. A straight line perpendicular to a coordinate plane in the term space is a 2-component zeroth-level element constant; e.g., line (Max, Bob, z) in Fig. 8.2. A plane perpendicular to a coordinate axis in the term space is a 1-component zeroth-level element constant; e.g., plane (Max, y, z) in Fig. 8.2.

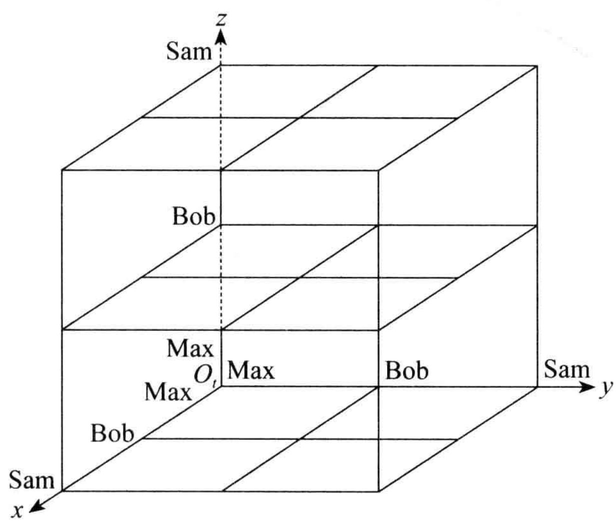


Fig. 8.2 Coordinate system of terms with empirical abstraction

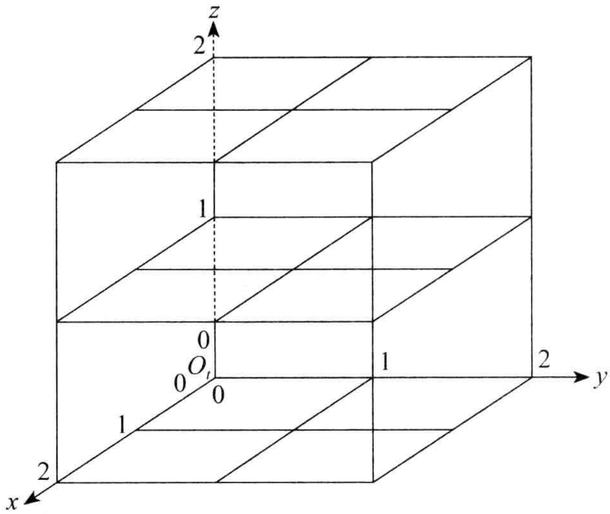


Fig. 8.3 Coordinate system of terms with mathematical abstraction

8.4.2 Coordinate system of facts

Coordinate systems of facts are shown in Figs. 8.4 to 8.6. In these figures, O_f is the origin. *P*−, *Q*−, *R*−axes are the coordinate axes of facts. *P*, *Q*, *R* are the zeroth-level set variables. The points on *P*−, *Q*−, *R*−axes are the zeroth-level set constants. The space formed by *P*−, *Q*−, *R*−axes is the fact space. A point in the fact space is a 3-component first-level element constant; e.g., point (x=y, y<z, x<z) in Fig. 8.5. A straight line perpendicular to a coordinate plane in the fact space is a 2-component first-level element constant; e.g., line (parent(x, y), grandparent(x, y), R) in Fig. 8.4

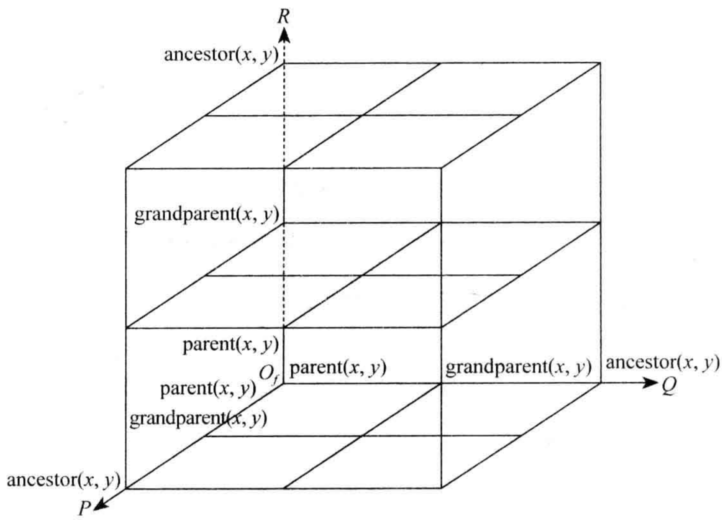


Fig. 8.4 Coordinate system of facts with empirical abstraction

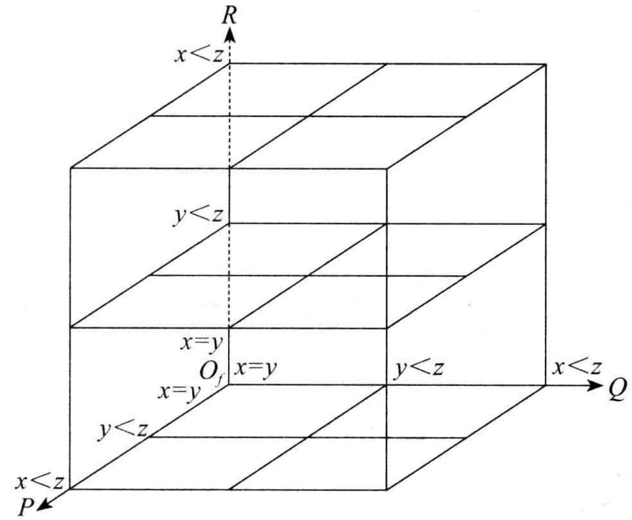


Fig. 8.5 Coordinate system of facts with mathematical abstraction

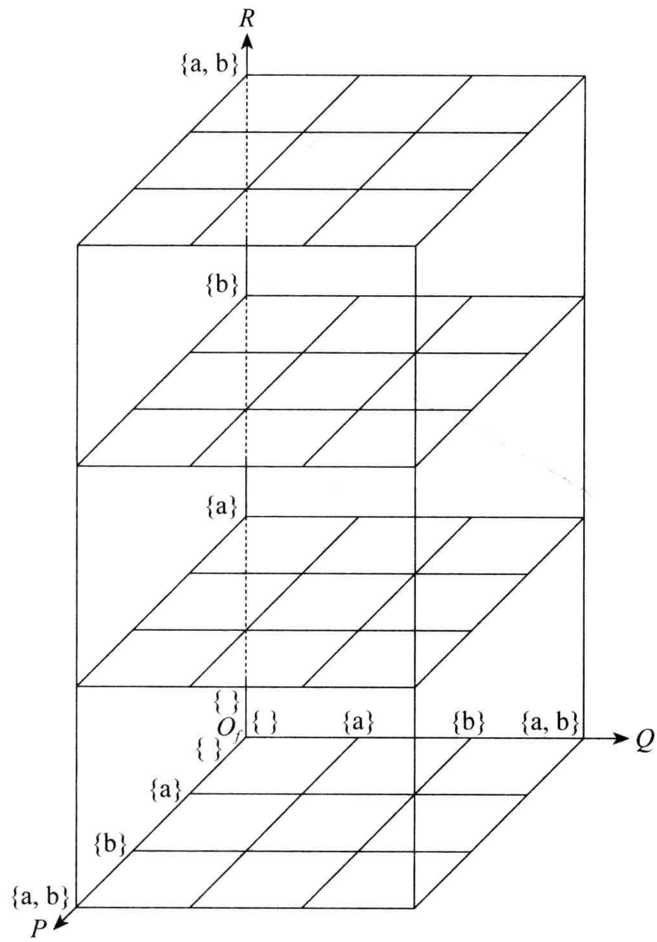


Fig. 8.6 Coordinate system of facts with set-theoretic abstraction

8.4.3 Relationship between the coordinate system of facts and the coordinate system of terms

The relationship between the coordinate system of facts and the coordinate system of terms is one between zeroth-level set constants and zeroth-level element constants. From the point of view of relational database, it is one between the type and value of a relation. A point on a coordinate axis of facts is a zeroth-level set constant (the type of a relation) containing such term variables as x, y, z, which form zeroth-level element variables varying through the x–, y–, z–axes in the term space. When x, y, z are assigned certain points in the x–, y–, z–axes, we obtain a zeroth-level element constant (the value of a relation) belonging or not to the zeroth-level set constant. All of the zeroth-level element constants belonging to the zeroth-level set constant constitute the set. For example, $x=y$ is a point on P-axis in Fig. 8.5, it is a zeroth-level set constant (the type of a relation), it contains the zeroth-level element variable (x, y) varying through the x–, y–axes in Fig. 8.3. When x is assigned 1, y 1, we obtain the zeroth-level element constant (1, 1); i.e., line $(1, 1, z)$ (the value of a relation), which belongs to $x=y$. (1, 1), (2, 2), (3, 3) constitute $x=y$. When x is assigned 1, y 2, we obtain (1, 2) not belonging to $x=y$.

8.4.4 Coordinate system of single empirical or mathematical connections

A coordinate system of single empirical or mathematical connection is shown in Fig. 8.7. In the figure, O_e is the origin. Ψ–, Ω–, Θ–axes are the coordinate axes of single empirical or mathematical connections. Ψ, Ω, and Θ are the first-level set variables. The points on Ψ-,

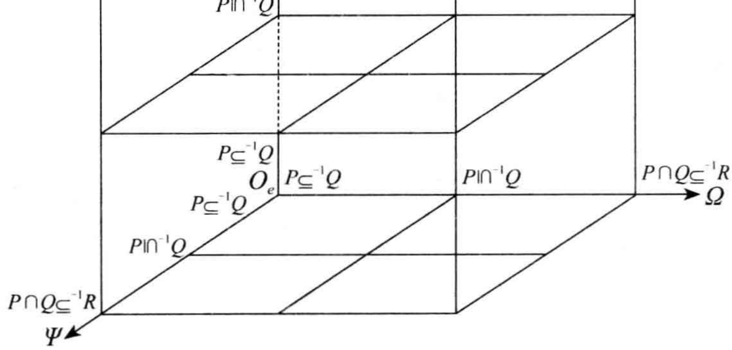


Fig. 8.7 Coordinate system of single empirical or mathematical connections

Ω-, Θ-axes are the first-level set constants. The space formed by Ψ–, Ω–, Θ–axes is the single empirical or mathematical connection space. A point in the single empirical or mathematical connection space is a 3-component second-level element constant; e.g., point $(P|\cap^{-1}Q, Q\subseteq^{-1}R, P|\cap^{-1}R)$. A straight line perpendicular to a coordinate plane in the single empirical or mathematical connection space is a 2-component second-level element constant; e.g., $(P\subseteq^{-1}Q, P|\cap^{-1}Q)$ in Fig. 8.7.

8.4.5 Relationship between the coordinate system of single empirical or mathematical connections and the coordinate system of facts

The relationship between the coordinate system of single empirical or mathematical connections and the coordinate system of facts is one between first-level set constants and first-level element constants. A point on a coordinate system of single empirical or mathematical connections is a first-level set constant containing such fact variables as P, Q, and R, which form first-level element variables varying through the P–, Q–, and R- axes. When P, Q, and R are assigned certain points in the P–, Q–, and R-axes, we obtain a first-level element constant belonging or not to the first-level set constant. All of the first-level element constants belonging to the first-level set constant constitute the set. For example, $P\subseteq^{-1}Q$ is a point on the Ψ–axis in Fig. 8.7, it is a first-level set constant, it contains the first-level element variable (P, Q) varying through the P– and Q–axes in Fig. 8.4. When P is assigned parent(x, y), Q ancestor(x, y), we obtain the first-level element constant (parent(x, y), ancestor(x, y)); i.e., line (parent(x, y), ancestor(x, y), R) belonging to $P\subseteq^{-1}Q$. (parent(x, y), ancestor(x, y)), (parent(x, y), parent(x, y)) constitute $P\subseteq^{-1}Q$. When P is assigned parent(x, y), Q grandparent(x, y), we obtain (parent(x, y), grandparent(x, y)) not belonging to $P\subseteq^{-1}Q$.

8.4.6 Coordinate system of single set connection

A coordinate system of single set connection is shown in Fig. 8.8. In the figure, O_s is the origin. Λ–, Δ–, and Ξ–axes are the coordinate axes of single set connections. Λ, Δ, and Ξ are the second-level set variables. The points on the Λ–, Δ–, and Ξ-axes are the second-level set constants.

8.4.7 Relationship between the coordinate system of single set connections and the coordinate system of single empirical or mathematical connections

The relationship between the coordinate system of single set connections and the coordinate system of single empirical or mathematical connections is one between second-level set constants and second-level element constants. A point on a coordinate axis of single set

connections is a second-level set constant containing such single empirical or mathematical connection variables as Ψ, Ω, and Θ, which form second-level element variables varying through the Ψ–, Ω–, and Θ–axes in the single empirical or mathematical connection space. When Ψ, Ω, and Θ are assigned certain points on the Ψ–, Ω–, and Θ–axes, we obtain a second-level element constant belonging or not to the second-level set constant. All of the second-level element constants belonging to the second-level set constant constitute the set. For example, $\Psi\cap\Omega\subseteq^{-1}\Theta$ is a point on the Λ-axis in Fig. 8.8, it is a second-level set constant, it contains the second-level element variable ($\Psi\cap\Omega$, Θ) varying through the Ψ–, Ω–, and Θ–axes in Fig. 8.7. When Ψ is assigned $P|\cap^{-1}Q$, Ω $Q\subseteq^{-1}R$, Θ $P|\cap^{-1}R$, we obtain the second-level element constant ($\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}$, $P|\cap^{-1}R$) belonging to $\Psi\cap\Omega\subseteq^{-1}\Theta$. ($\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}$, $P|\cap^{-1}R$) and ($\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}R\}$, $P\subseteq^{-1}R$) constitute $\Psi\cap\Omega\subseteq^{-1}\Theta$. When Ψ is assigned $P\subseteq^{-1}Q$, Ω $Q|\cap^{-1}R$, Θ $P|\cap^{-1}R$, we obtain ($\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}$, $P|\cap^{-1}R$) not belonging to $\Psi\cap\Omega\subseteq^{-1}\Theta$.

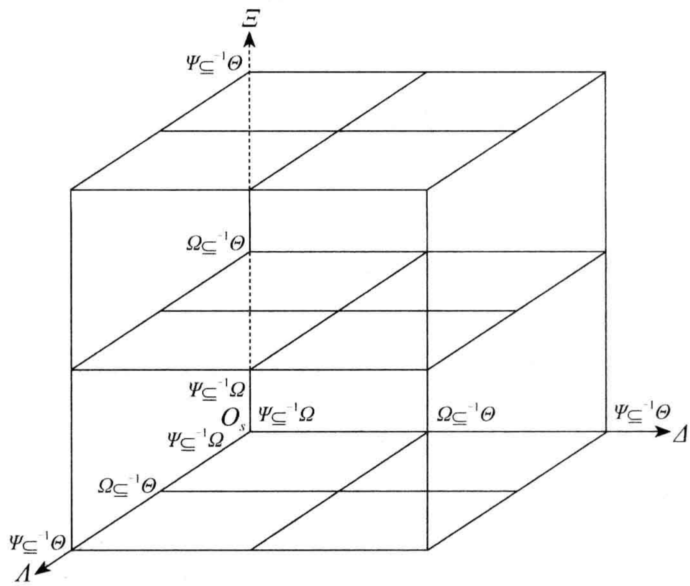


Fig. 8.8 Coordinate system of single set connections

8.5 Intersections

8.5.1 Fact intersections

8.5.1.1 Fact basic intersection

Suppose A={a, b}, B={b, c}, then the fact basic intersection of A and B is A∩B={b}. A is actually A(x), B is actually B(x). As both A(x) and B(x) have only one term variable, and both are x, we can omit it.

8.5.1.2 Fact extended intersection

If a set is one of n-tuples, then the type of the set (relation) has n term variables, at this time, the n variables cannot be omitted. For example, suppose A(x, y)={(a, b), (b, c)}. The type of the set A(x, y) has two term variables: x and y. The b in the first binary tuple (a, b) of the set is an instance of y, while the b in the second binary tuple (b, c) is an instance of x, therefore A(x, y) cannot be simplified as A. Intersection operations are developed for relational database, therefore we denote sets as tables. A(x, y)={(a, b), (b, c)} is denoted as Table 8.6. B(x, y)={(a, b), (a, c)} as Table 8.7.

Table 8.6 A(x, y)

x	y
a	b
b	c

Table 8.7 B(x, y)

x	y
a	b
a	c

Table 8.8 A(x, y)$\cap$B(x, y)

x	y
a	b

As the corresponding term variables of A(x, y) and B(x, y) are identical, fact extended intersection A(x, y)$\cap$B(x, y) can be performed on them, as is shown in Table 8.8.

8.5.1.3 Fact first intersection

Suppose there is the first-order single empirical or mathematical connection proposition $x>z\cap y>z\subseteq^{-1}x+y>z$, and we want to make an intersection operation on $x>z$ and $y>z$. As the first term variables of $x>z$ and $y>z$ are different, fact extended intersection cannot be performed on them. Fact first intersection is defined as follows: in $x>z\cap y>z\subseteq^{-1}x+y>z$, x and y are relevantly bound term variables, z is a relevantly and intermediarily bound term variable, from the perspective of relational database, natural join is made on the z of table $x>z$ and table $y>z$, then fact first intersection of $x>z$ and $y>z$ is obtained.

Suppose $x>z$ is given in Table 8.9, $y>z$ in Table 8.10, then $x>z\cap y>z$ is shown in Table 8.11, from which we learn that $x+y>z$ holds.

Table 8.9 $x>y$

x	z
1	0
2	0
2	1

Table 8.10 $y>z$

y	z
1	0
2	0
2	1

Table 8.11 $x>z\cap y>z$

x	y	z
1	1	0
2	1	0
1	2	0
2	2	0
2	2	1

8.5.1.4 Fact second intersection

Suppose there is the first-order single empirical or mathematical connection proposition parent(x, y)$\cap$ancestor(y, z)$\subseteq^{-1}$ancestor(x, z), and we want to make an intersection operation on parent(x, y) and ancestor(y, z). As the corresponding term variables of parent(x, y) and

ancestor(y, z) are all different, fact extended intersection cannot be performed on them. Fact second intersection is defined as follows: in parent(x, y)$\cap$ancestor(y, z)$\subseteq^{-1}$ancestor(x, z), x and z are relevantly bound term variables, y is a intermediarily bound term variable, from the perspective of relational database, natural join is made on y to tables parent(x, y) and ancestor(y, z), then projection is made on x and z, the table finally obtained is the fact second intersection of parent(x, y) and ancestor(y, z).

Suppose parent(x, y) is given in Table 8.12, ancestor(y, z) in Table 8.13, then natural join on y to Tables 8.12 and 8.13 is shown in Table 8.14, and projection on x and z to Table 8.14 is shown in Table 8.15, from which we learn that ancestor(x, z) holds.

Table 8.12 Parent(x, y)

x	y
Bob	Max
Max	Sam

Table 8.13 Ancestor(y, z)

y	z
Bob	Max
Max	Sam

Table 8.14 Natural join on y

x	y	z
Bob	Max	Sam

Table 8.15 Projection on x and z

x	z
Bob	Sam

8.5.2 Empirical or mathematical intersections

8.5.2.1 Empirical or mathematical extended intersection

Suppose there is second-order single set connection proposition $\{P\subseteq^{-1}Q\}\cap\{P\supseteq^{-1}Q\}\subseteq^{-1}$ $\{P=^{-1}Q\}$ in which $\supseteq^{-1}$ denotes mutually inversely containing. We want to make intersection operation on $P\subseteq^{-1}Q$ and $P\supseteq^{-1}Q$. As the corresponding fact variables of $P\subseteq^{-1}Q$ and $P\supseteq^{-1}Q$ are identical, empirical or mathematical extended intersection can be made on them. Suppose $P\subseteq^{-1}Q$ is given in Table 8.16, $P\supseteq^{-1}Q$ in Table 8.17, then the empirical or mathematical extended intersection of $P\subseteq^{-1}Q$ and $P\supseteq^{-1}Q$ is shown in Table 8. 18, from which we learn that $P=^{-1}Q$ holds.

Table 8.16 $P\subseteq^{-1}Q$

P	Q
equilateral_triangle(x)	equiangular_triangle(x)
equilateral_triangle(x)	isosceles_triangle(x)

Table 8.17 $P\supseteq^{-1}Q$

P	Q
equilateral_triangle(x)	equiangular_triangle(x)
isosceles_triangle(x)	equilateral_triangle(x)

Table 8.18 $\{P\subseteq^{-1}Q\}\cap\{P\supseteq^{-1}Q\}$

P	Q
equilateral_triangle(x)	equiangular_triangle(x)

8.5.2.2 Empirical or mathematical first intersection

Suppose there is second-order single set connection proposition $\{P\subseteq^{-1}Q\}\cap\{P\subseteq^{-1}R\}\subseteq^{-1}\{P\subseteq^{-1}Q\cap R\}$. We want to make intersection operation on $P\subseteq^{-1}Q$ and $P\subseteq^{-1}R$. As the second fact variables of $P\subseteq^{-1}Q$ and $P\subseteq^{-1}R$ are different, empirical or mathematical extended intersection cannot be made. Empirical or mathematical first intersection is defined as follows: in $\{P\subseteq^{-1}Q\}\cap\{P\subseteq^{-1}R\}\subseteq^{-1}\{P\subseteq^{-1}Q\cap R\}$, Q and R are relevantly bound fact variables, P is a relevantly and intermediarily bound fact variable, from the perspective of second-level single quasi-relational database, a natural join is made on P to $P\subseteq^{-1}Q$ and $P\subseteq^{-1}R$, then $\{P\subseteq^{-1}Q\}\cap\{P\subseteq^{-1}R\}$ is obtained.

Suppose $P\subseteq^{-1}Q$ is given in Table 8.19, $P\subseteq^{-1}R$ in Table 8.20, then $\{P\subseteq^{-1}Q\}\cap\{P\subseteq^{-1}R\}$ is shown in Table 8.21, from which we learn that $P\subseteq^{-1}Q\cap R$ holds.

Table 8.19 $P\subseteq^{-1}Q$

P	Q
neg_int(x)	neg(x)
pos_rat(x)	pos(x)

Table 8.20 $P\subseteq^{-1}R$

P	R
neg_int(x)	int(x)
pos_rat(x)	rat(x)

Table 8.21 $\{P\subseteq^{-1}Q\}\cap\{P\subseteq^{-1}R\}$

P	Q	R
neg_int(x)	neg(x)	int(x)
pos_rat(x)	pos(x)	rat(x)

8.5.2.3 Empirical or mathematical second intersection

Suppose there is second-order single set connection proposition $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P\subseteq^{-1}R\}$. We want to make intersection operation to $P\subseteq^{-1}Q$ and $Q\subseteq^{-1}R$. As the corresponding fact variables of $P\subseteq^{-1}Q$ and $Q\subseteq^{-1}R$ are all different, empirical or mathematical extended intersection cannot be made. Empirical or mathematical second intersection is defined as follows: in $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P\subseteq^{-1}R\}$, P and R are relevantly bound fact variables, Q is an intermediarily bound fact variable, from the perspective of second-level single quasi-relational database, a natural join is made on Q to $P\subseteq^{-1}Q$ and $Q\subseteq^{-1}R$, and then for the table obtained, a projection is made on P and R, obtaining $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}R\}$.

Suppose $P\subseteq^{-1}Q$ is given in Table 8. 22, $Q\subseteq^{-1}R$ in Table 8.23, then a natural join on Q to Tables 8.22 and 8.23 is performed, obtaining Table 8.24, finally, a projection on P and R to Table 8.24 is performed, obtaining Table 8.25, from which we learn that $P\subseteq^{-1}R$ holds.

Table 8.22 $P\subseteq^{-1}Q$

P	Q
int(x)	rat(x)
rat(x)	real(x)

Table 8.23 $Q\subseteq^{-1}R$

P	Q
int(x)	rat(x)
rat(x)	real(x)

Table 8.24 Natural join on *Q*

P	*Q*	*R*
int(x)	rat(x)	real(x)

Table 8.25 Projection on *P* and *R*

P	*E*
int(x)	real(x)

8.6 Power sets

Suppose A is a set, then the set composed of all the subsets of A is called the power set of A, denoted by ρ(A). For example, suppose A={a, b}, then ρ(A)={∅, {a}, {b}, {a, b}}.

8.7 Principle of meaningfulness—meaninglessness duality for distinguished sets

An empty set does not reflect objective being, there is nothing in the empty set. For example, dragon(x) is an empty set, for there is no dragon in the universe. A universal set is an abstraction of the universe, it is ubiquitous. An empty set and a universal set are distinguished sets. But, in the abstract set operations, distinguished sets are bound to be produced, to be operated on. For example, the intersection of P and $\sim P$ is an empty set.

The principle of meaningfulness—meaninglessness duality for distinguished sets: distinguished sets occur in quasi-set connection propositions with the outmost connection operators being$=^{-1}$, in quasi-transcendent set connection propositions with the outmost connection operators being$=^{-1}$, in power sets are meaningful ones; distinguished sets occur elsewhere are meaningless ones.

8.8 Paradoxes

In mutually-inversistic set theory, no logical or mathematical paradox holds.

8.8.1 Russell's paradox

Russell set is the set of all non-self-membered sets, denoted as:

$$\mathrm{R}=\{x|x\notin x\} \tag{8.1}$$

Or

$$x\in\mathrm{R}\leftrightarrow x\notin x \tag{8.2}$$

When x is assigned R, formula (8.2) becomes:

$$\mathrm{R}\in\mathrm{R}\leftrightarrow\mathrm{R}\notin\mathrm{R} \tag{8.3}$$

From formula (8.3), we learn that R∈R and R∉R are mutually deducible. This is the famous Russell's paradox.

In mutually-inversistic set theory, Russell's paradox does not hold, which can be

viewed from two angles. First, from the angle of levels of elements and sets, x is a zeroth-level element variable, varying through a coordinate axis of terms, while R is a zeroth-level set constant, it is a point on a coordinate axis of facts. X cannot be assigned R. Secondly, from the angle of the principle of meaningfulness—meaninglessness duality for distinguished sets, $x \notin x$ is a universal set, a meaningless distinguished set, while mutually-inversistic set theory only studies meaningful sets.

8.8.2 The greatest ordinal number paradox

Suppose O_n is a set composed of all ordinal numbers. $O_n^+ = O_n \cup \{O_n\}$ is also an ordinal number, and we have $O_n \in O_n^+$. However, because O_n is a set composed of all ordinal numbers, we also have $O_n^+ \in O_n$. This is the greatest ordinal number paradox.

In mutually-inversistic set theory, the greatest ordinal number paradox does not hold: O_n is a universal set, a meaningless distinguished set, while mutually-inversistic set theory only studies meaningful sets.

8.8.3 The greatest cardinal number paradox

Cantor's theorem: the cardinality $|M|$ of any set M is less than the cardinality $|\rho(M)|$ of its power set $\rho(M)$.

Suppose set U is composed of all sets. $|U| < |\rho(U)|$ is obtained by Cantor's theorem. But U is a set of all sets, therefore, U should contain $\rho(U)$; i.e., $|U| \geqslant |\rho(U)|$. Thus, both $|U| < |\rho(U)|$ and $|U| \geqslant |\rho(U)|$ hold. This is the greatest cardinal number paradox.

In mutually-inversistic set theory, the greatest cardinal number paradox does not hold: U is a universal set, a meaningless distinguished set, while mutually-inversistic set theory only studies meaningful sets.

8.9 Ordinal numbers and cardinal numbers are wrong theories

In naïve set theory and axiomatic set theory, natural numbers are constructed from empty sets. 0, 1, 2, 3,… are denoted by $\varnothing$, $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$, $\{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$, … , respectively. The construction method is as follows:

$0=\varnothing$,

$x'=x \cup \{x\}$ (x' denotes the successor of x).

Thus, we have:

$0=\varnothing$,

$1=0'=0 \cup \{0\}=\varnothing \cup \{\varnothing\}=\{\varnothing\}=\{0\}$,

$2=1'=1 \cup \{1\}=\{\varnothing\} \cup \{\{\varnothing\}\}=\{\varnothing, \{\varnothing\}\}=\{0, 1\}$,

$3=2'=2\cup\{2\}=\{\varnothing, \{\varnothing\}\}\cup\{\{\varnothing, \{\varnothing\}\}\}=\{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}=\{0, 1, 2\}$,

According to the above construction, we have both:

$$0\subset 1\subset 2\subset 3\subset \ldots \tag{8.4}$$

and

$$0\in 1\in 2\in 3\in \ldots \tag{8.5}$$

Take 2 and 3 as an example, we have both $2\subset 3$ (there are two elements in 2: $\varnothing$ and $\{\varnothing\}$, they are both elements of 3); and $2\in 3$ (set $2=\{\varnothing, \{\varnothing\}\}$, which is an element of 3).

Formula (8.5) is wrong. Natural numbers are mathematical abstractions of the nature. For example, natural number 3 is a mathematical abstraction of 3 people, 3 trees, 3 houses, ... The set of 2 people {Frege, Cantor} can only be a subset of the set of 3 people {Frege, Cantor, Hilbert}; i.e., {Frege, Cantor}$\subset${Frege, Cantor, Hilbert}. The former cannot belongs to the latter; i.e., we cannot have {Frege, Cantor}$\in${Frege, Cantor, Hilbert}. Therefore, (8.5) is wrong. That (8.5) is wrong does not mean that natural numbers are wrong theory, it only means that natural numbers cannot be constructed this way.

Ordinal number theory is a generalization of (8.5). An ordinal number is defined as a transitive set of $\in$_well-ordering, where $\in$_well-ordering is (8.5). Therefore, ordinal number theory is a wrong theory. A cardinal number is defined as the smallest of equipollent ordinal numbers, therefore, cardinal number theory is also a wrong theory. In continuum hypothesis: $\aleph_1=2^{\aleph_0}$, where $\aleph_0$ is the cardinal number of natural numbers, $\aleph_1$ is the cardinal number of real numbers. Continuum hypothesis is a wrong problem, because cardinal numbers are wrong theory.

In mutually-inversistic set theory, these wrong theories cannot occur, because $\varnothing$s in the above construction are meaningless distinguished sets, the construction is meaningless.

8.10 Comparison of mutually-inversistic set theory with naïve set theory and axiomatic set theory

In naïve set theory, there are paradoxes. Axiomatic set theory is paradox-free, but is complicated. It has many axioms and rules, it distinguishes sets from classes. Mutually-inversistic set theory is paradox-free because of two principles: first, elements and sets are divided into three levels, secondly, meaningless distinguished sets are disallowed. That elements and sets are divided into three levels is an improvement of type theory. Meaningless distinguished sets are just proper classes. In mutually-inversistic set theory there is no need to distinguish sets from classes, and proper classes are confined within empty sets and universal sets. Mutually-inversistic set theory is simpler than axiomatic set theory.

Chapter 9
The Main

This chapter studies the second-order main constituents; i.e., binary relations and the third-order main constituents; i.e., empirical or mathematical connection operators.

9.1 Binary relations

A binary relation, a relation for short, is a two-place predicate.

9.1.1 Digraphs

A binary relation can be represented pictorially either by a mutually inverse diagram in a term space or by a directed graph (digraph). The digraph of relation R on the universe of terms I_0 is constructed as follows. Draw a small circle and label it with a_i to represent the term constant $a_i \in I_0$. The circles are called vertices. If $(a_i, a_j) \in R$, then draw an arrow, called an edge, from vertex a_i to vertex a_j. If $(a_i, a_i) \in R$, then draw an arrow, called a cycle, from vertex a_i to itself.

Example 9.1: Let $I_0=\{1, 2, 3, 4, 5\}$, $r_1(x, y)=\{(1, 2), (2, 2), (3, 2), (3, 4), (4, 3)\}$. Then the digraph of $r_1(x, y)$ is shown in Fig. 9.1.

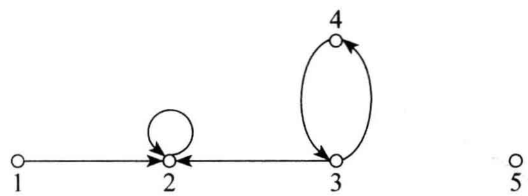


Fig. 9.1 Digraph

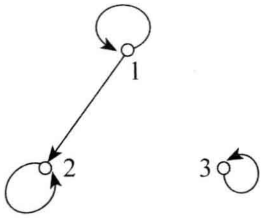
9.1.2 Properties of relations

Definition 9.1: Suppose r is a binary relation on I_0. If for every $x \in I_0$, $r(x, x)$ holds, then r is reflexive.

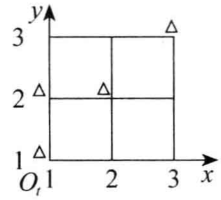
Example 9.2: Suppose $I_0=\{1, 2, 3\}$, $r_2(x, y)=\{(1, 1), (2, 2), (3, 3), (1, 2)\}$. R_2 is reflexive, its digraph and mutually inverse diagram are shown in Fig. 9.2.

Definition 9.2: Suppose r is a binary relation on I_0. If for every $x \in I_0$, $\sim r(x, x)$ holds, then r is irreflexive.

Example 9.3: Suppose $I_0=\{1, 2, 3\}$, $r_3(x, y)=\{(2, 1), (1, 3), (3, 2)\}$. R_3 is irreflexive, its digraph and mutually inverse diagram are shown in Fig. 9.3.

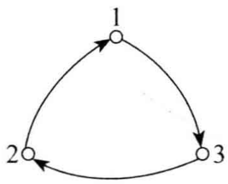


Every vertex has a cycle

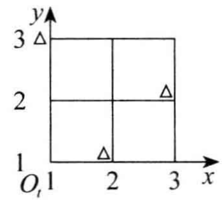


All elements on the main diagonal belong to the mutually inverse diagram

Fig. 9.2 Pictorial representations of reflexive relation



No vertex has a cycle

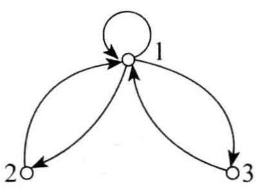


No element on the main diagonal belongs to the mutually inverse diagram

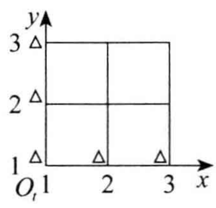
Fig. 9.3 Pictorial representation of irreflexive relation

Definition 9.3: Suppose r is a binary relation on I_0, if for every $x, y \in I_0$, $r(x, y) \overset{-1}{\subseteq} r(y, x)$ holds, then r is symmetric.

Example 9.4: Suppose $I_0=\{1, 2, 3\}$, $r_4(x, y)=\{(1, 2), (2, 1), (1, 3), (3, 1), (1, 1)\}$. R_4 is symmetric, its digraph and mutually inverse diagram are shown in Fig. 9.4.



If there is an edge from a to b, then there is bound to be an edge from b to a



Elements in the mutually inverse diagram are symmetric with regard to the main diagonal

Fig. 9.4 Pictorial representation of symmetric relation

Definition 9.4: Suppose r is a binary relation on I_0, if for every $x, y \in I_0$, $r(x, y) \cap x \neq y \overset{-1}{\subseteq} \sim r(y, x)$ holds, then r is antisymmetric.

Example 9.5: Suppose $I_0=\{1, 2, 3\}$, $r_5(x, y)=\{(1, 2), (2, 3)\}$. R_5 is antisymmetric, its digraph and mutually inverse diagram are shown in Fig. 9.5.

If there is an edge from a to b, then there is no edge from b to a

If a_{ji} belongs to mutually inverse diagram, then a_{ij} does not belong to it, here $i \neq j$

Fig. 9.5 Pictorial representation of antisymmetric relation

Definition 9.5: Suppose r is a binary relation on I_0, if for every $x, y, z \in I_0$, $r(x, y) \cap r(y, z) \subseteq^{-1} r(x, z)$ holds, then r is transitive.

Example 9.6: Suppose $I_0=\{1, 2, 3\}$, $r_6(x, y)=r_6(y, z)=r_6(x, z)=\{(1, 2), (2, 3), (1, 3)\}$. R_6 is transitive, its digraph is shown in Fig. 9.6, its mutually inverse diagram in Fig. 9.7.

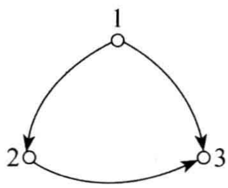


Fig. 9.6 Digraph for transitive relation

The characteristic of Fig. 9.6 is that, if there exists a path from a to b, then there exists an edge from a to b. The path from a to b refers to a sequence of vertices $a=a_0, a_1, a_2, \ldots, a_n=b$, for every i $(0 \leqslant i \leqslant n-1)$, there exists an edge from a_i to a_{i+1}.

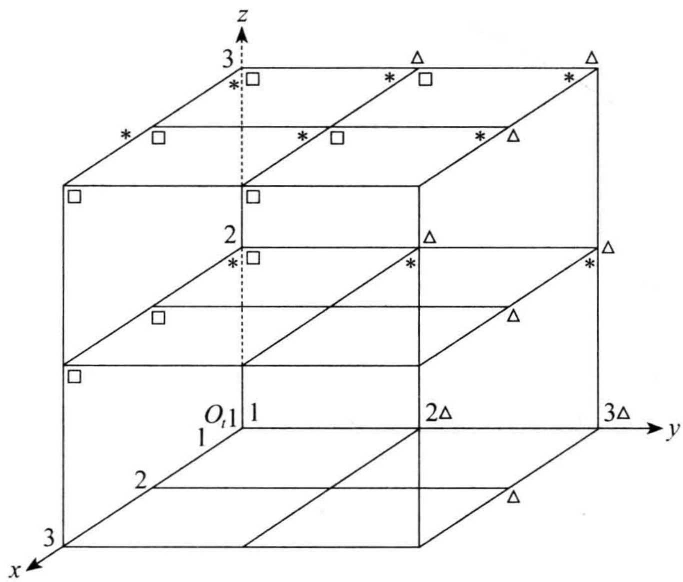


Fig. 9.7 Mutually inverse diagram for transitive relation

In Fig. 9.7, $r_6(x, y)$ is denoted by vertices marked with "△", $r_6(y, z)$ by "□", $r_6(x, z)$ by "∗". $R_6(x, y) \cap r_6(y, z)$ is vertex (1, 2, 3) (the vertex is marked with both "△" and "□"), the vertex also belongs to $r_6(x, z)$ (it is also marked with "∗"). That is, $r_6(x, y) \cap r_6(y, z) \subseteq^{-1} r_6(x, z)$ holds.

In naïve set theory, relational matrices are used to denote binary relations. But relational matrices cannot denote transitivity. In mutually-inversistic set theory, mutually inverse diagrams in term spaces are used to denote binary relations. From Fig. 9.7, we see that mutually inverse diagrams can denote transitivity. This is one advantage of mutually inverse diagrams over relational matrices.

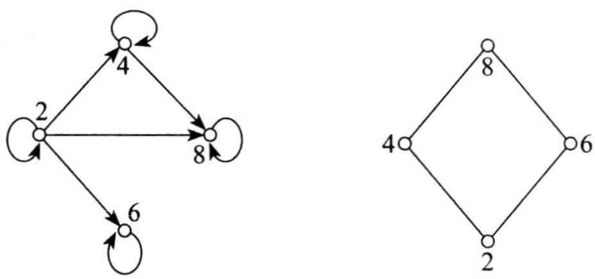
9.1.3 Order relations

9.1.3.1 Partially ordered sets

Definition 9.6: If a binary relation r on I_0 is reflexive, antisymmetric, and transitive, then r is called a partial order on I_0, and the ordered pair $\langle I_0, r\rangle$ is called a partially ordered sets.

If r is a partial order, then $\langle I_0, r\rangle$ is usually denoted as $\langle I_0, \leqslant\rangle$, because $\leqslant$ is also a partial order, and $r(a, b)$ is denoted as $a \leqslant b$.

Example 9.7: Suppose $I_0=\{2, 4, 6, 8\}$, div denotes the relation of divisibility, then $div(x, y)=\{(2, 2), (4, 4), (6, 6), (8, 8), (2, 4), (2, 6), (2, 8), (4, 8)\}$. $\langle I_0, div\rangle$ is a partially ordered set. Its digraph is shown in Fig. 9.8 (a). A partially ordered set is usually denoted by a Hasse diagram. In the Hasse diagram, the cycles are omitted. And the edges implied by transitive property are omitted. For example, if $x \leqslant y$ and $y \leqslant z$, it follows that $x \leqslant z$, and the edge of $x \leqslant z$ is omitted. All edges point upward, so that arrows may be omitted from the edges. The Hasse diagram of this example is shown in Fig. 9.8 (b).



(a) (b)

Fig. 9.8 Pictorial representation of partial order

Definition 9.7: Suppose $\langle I_0, \leqslant\rangle$ is a partially ordered set, I_0' is a subset of I_0.

(1) If $a \in I_0'$, and there is no element $x \in I_0'$ such that $a \neq x$ and $a \leqslant x$, then a is called a maximal element of I_0'.

(2) If $a \in I_0$', and there is no element $x \in I_0$' such that $a \neq x$ and $x \leqslant a$, then a is called a minimal element of I_0'.

(3) Element $a \in I_0$' is the greatest element of I_0', if for every element $x \in I_0$', $x \leqslant a$.

(4) Element $a \in I_0$' is the least element of I_0', if for every element $x \in I_0$', $a \leqslant x$.

(5) If for every element $b \in I_0$', $b \leqslant a$, then element $a \in I_0$ is called an upper bound of I_0'; if for every element $b \in I_0$', $a \leqslant b$, then element $a \in I_0$ is called the lower bound of I_0'.

(6) If a is an upper bound of I_0', and for every upper bound a' of I_0', $a \leqslant a'$, then $a \in I_0$ is called a least upper bound (lub) of I_0'; if a is a lower bound of I_0', and for every lower bound a' of I_0', $a' \leqslant a$, then $a \in I_0$ is called a greatest lower bound (glb) of I_0'.

Example 9.8: Suppose the partially ordered set is <{2, 5, 6, 10, 15, 30}, div>, its Hasse diagram is shown in Fig. 9.9. Suppose I_0'is {2, 5, 6, 10, 15, 30}, then 2 and 5 are the minimal elements of I_0', but there is no least element, lower bound, and greatest lower bound in I_0'. Element 30 is the maximal element, greatest element, upper bound, and least upper bound of I_0'

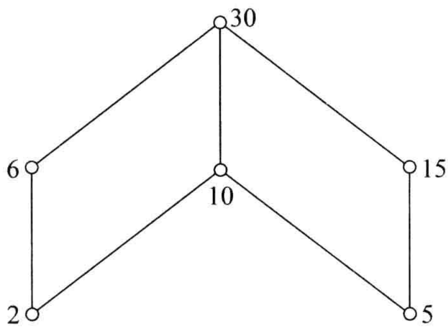


Fig. 9.9 An example of partial orders

9.1.3.2 Quasi-ordered sets

Definition 9.8: If a binary relation r on I_0 is irreflexive, antisymmetric, and transitive, then r is called a quasi-order on I_0, $\langle I_0, r\rangle$ is called a quasi-ordered set. Usually, quasi-order is denoted by $<$.

Example 9.9: $<$ on the set of real numbers **R** is a quasi-order.

9.1.3.3 Linearly ordered sets and well-ordered sets

If $\leqslant$ is a partial order, and either $a \leqslant b$ or $b \leqslant a$ hold, then we say that a and b are comparable. Elements in a partially ordered set may not be comparable.

Definition 9.9: In a partially ordered set $\langle I_0, \leqslant\rangle$, if for every a, $b \in I_0$, either $a \leqslant b$ or $b \leqslant a$, then $\leqslant$ is called a linear order (or total order)on I_0, $\langle I_0, \leqslant\rangle$ is called a linearly ordered set.

Example 9.10: $\langle \mathbf{Z}, \leqslant\rangle$ is a linearly ordered set, where **Z** denotes the set of integers.

Definition 9.10: If a binary relation r on I_0 is a linear order, and every non-empty set of I_0 has a least element, then r is called a well-order on I_0, $\langle I_0, r\rangle$ is called a well-ordered set.

Example 9.11: $\langle \mathbf{Z}, \leqslant\rangle$ is not a well-ordered set, because some subsets of **Z**; e.g., **Z** itself, do not have least elements.

9.1.4 Equivalence relations and partitions

Definition 9.11: If a binary relation r on I_0 is reflexive, symmetric, and transitive, then r is called an equivalence relation.

Example 9.12: "living in the same room", "the same age" are all equivalence relations. "classmate" is not an equivalence relation.

Example 9.13: Equality relation = is an equivalence relation on numbers.

Modular equivalence is an equivalence relation on integers or its subsets.

Definition 9.12: Suppose k is an integer, a, b$\in$**Z**. If for certain integer m, a–b=m$*$k, then a and b are equivalent modulo k, written as:

$a \equiv b \pmod{k}$,

and the integer k is called the module of the equivalence.

Example 9.14: Suppose r is an equivalence relation mod 4 on **Z**, then

$[0]_4=\{\ldots, -8, -4, 0, 4, 8,\ldots\}$,

$[1]_4=\{\ldots, -7, -3, 1, 5, 9,\ldots\}$,

$[2]_4=\{\ldots, -6, -2, 2, 6, 10,\ldots\}$,

$[3]_4=\{\ldots, -5, -1, 3, 7, 11,\ldots\}$.

The numbers in each set are mutually equivalent.

Definition 9.13: Suppose r is an equivalence relation on I_0, for every a$\in I_0$, the equivalence class of a with respect to r is the set $\{x|r(x, a)\}$, denoted as $[a]_r$, a is called the representative of the equivalence class $[a]_r$.

Example 9.15: The equivalence classes of the equivalence relation modulo 4 on **Z** are $[0]_4$, $[1]_4$, $[2]_4$, $[3]_4$.

Definition 9.14: Given a non-empty set A and a non-empty set family $\pi=\{A_1, A_2, \ldots, A_m\}$, if $A=^{-1}A_1\cup A_2\cup\ldots\cup A_m$, then the set family π is called a covering of A.

Definition 9.15: Given a non-empty set A and a non-empty set family $\pi=\{A_1, A_2, \ldots, A_m\}$, if

(1) π is a covering of A,

(2) Either $\sim\{A_i|\cap^{-1}A_j\}$; i.e., A_i and A_j do not intersect each other, or $A_i=^{-1}A_j$ (i, j=1, 2,…,m),

then the set family π is called a partition of the set A.

Example 9.16: The set of the equivalence classes $\{[0]_4, [1]_4, [2]_4, [3]_4\}$ is a partition of **Z**.

9.1.5 Compatible relations and coverings

Definition 9.16: Suppose r is a binary relation on I_0, if r is reflexive and symmetric, then r is called a compatible relation.

Example 9.17: Suppose I_0={316, 347, 204, 678, 770}, $r_7(x, y)$ is that x and y have the same digit. R_7 is a compatible relation, its digraph is shown in Fig. 9.10, where a=316, b=347, c=204, d=678, and e=770

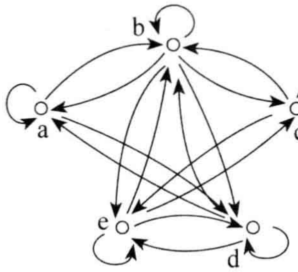


Fig. 9.10 Digraph for compatible relation

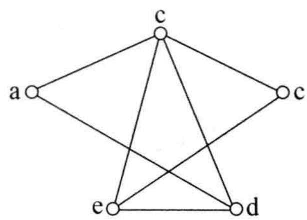


Fig. 9.11 Pictorial representation of compatible relation

Fig. 9.10 can be simplified to Fig. 9.11 by omitting the cycles and using undirected edges to replace two directed edges.

Definition 9.17: Suppose r is a compatible relation on A, and B is a subset of A. If every $x \in$B has compatible relation with all other elements in B, and no element in A−B has compatible relation with all elements in B, then subset B is called a maximal compatible class of the compatible relation r.

In Example 9.17, A_1={a, b, d}, A_2={b, c, e}, A_3=b, d, e} are all maximal compatible classes of r_7.

The set of all the maximal compatible classes of a compatible relation r on A is a covering of A.

9.2 Empirical or mathematical connection operators

Suppose I_1 is a universe of fact propositions, φ is an empirical or mathematical connection operator on I_1.

Definition 9.18:

If $P\varphi P$ holds, then φ is reflexive.

If $\sim\{P\varphi P\}$ holds, then φ is irreflexive.

If $\{P\varphi Q\}\subseteq^{-1}\{Q\varphi P\}$ holds, then φ is symmetric.

If $\{P\varphi Q\}\cap\{P\neq^{-1}Q\}\subseteq^{-1}\sim\{Q\varphi P\}$, then φ is antisymmetric.

If $\{P\varphi Q\}\cap\{Q\varphi R\}\subseteq^{-1}\{P\varphi R\}$, then φ is transitive.

9.2.2 A partially ordered connection operator

Definition 9.19: If an empirical or mathematical connection operator φ on I_1 is reflexive, antisymmetric, and transitive, then φ is called a partially ordered connection operator on I_1.

Example 9.18: Suppose $I_1 =^{-1} \{\text{integer}(x), \text{rational}(x), \text{real\_number}(x)\}$, then $\subseteq^{-1}$ is a partially ordered connection operator on I_1, its mutually inverse diagram is shown in Fig. 9.12

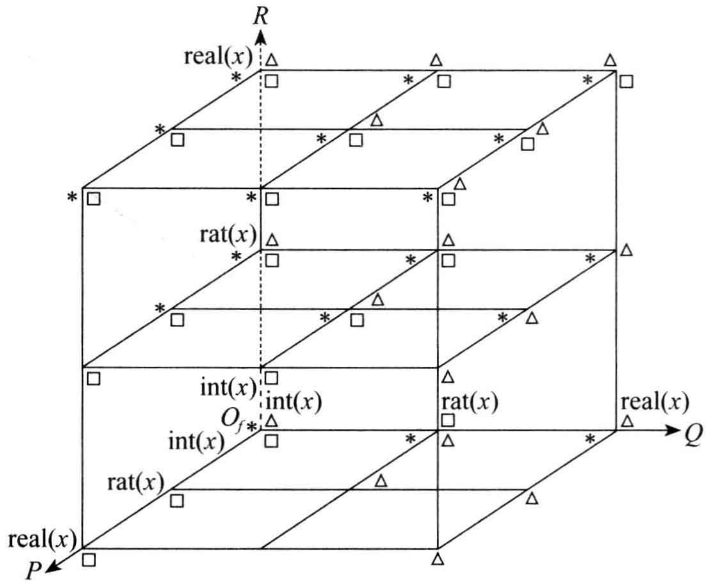


Fig. 9.12 Mutually inverse diagram for partially ordered connection operator

Reflexivity of $\subseteq^{-1}$ is reflected by the PO_fQ plane of Fig. 9.12, where elements on the main diagonal are all marked with "△". Antisymmetry of $\subseteq^{-1}$ is reflected by the PO_fQ plane, where element (int(x), real(x)) is marked with "△", while element (real(x), int(x)) is not. Transitivity of $\subseteq^{-1}$ is reflected by Fig. 9.12, where $P \subseteq^{-1} Q$ is denoted by vertices marked with "△", $Q \subseteq^{-1} R$ is denoted by vertices marked with "□", $P \subseteq^{-1} R$ is denoted by vertices marked with "∗", the vertices marked with both "△" and "□" are bound to be marked with "∗"; i.e., $\{P \subseteq^{-1} Q\} \cap \{Q \subseteq^{-1} R\} \subseteq^{-1} \{P \subseteq^{-1} R\}$ holds.

9.2.3 A quasi-ordered connection operator

Definition 9.20: If an empirical or mathematical connection operator φ on I_1 is irreflexive, antisymmetric, and transitive, then φ is called a quasi-ordered connection operator on I_1.

Example 9.19: Suppose $I_1 =^{-1} \{\text{integer}(x), \text{rational}(x), \text{real\_number}(x)\}$, then $\subset^{-1}$ is a quasi-ordered connection operator on I_1.

9.2.4 An equivalent connection operator

Definition 9.21: If an empirical or mathematical connection operator φ on I_l is reflexive, symmetric, and transitive, then φ is an equivalent connection operator on I_l.

Example 9.20: Suppose $I_l =^{-1} \{\text{integer}(x), \text{rational}(x), \text{real\_number}(x)\}$, then $=^{-1}$ is an equivalent connection operator on I_l.

9.2.5 A compatible connection operator

Definition 9.22: If an empirical or mathematical connection operator φ on I_l is reflexive, and symmetric, then φ is a compatible connection operator on I_l.

Example 9.21: Suppose $I_l =^{-1} \{\text{parent}(x, y), \text{grandparent}(x, y), \text{ancestor}(x, y)\}$, then $|\cap^{-1}$ is a compatible connection operator on I_l, its mutually inverse diagram is shown in Fig. 9.13, where all the elements on the main diagonal are marked with "△"; i.e., $|\cap^{-1}$ is reflexive, and the elements marked with "△"are symmetric with respect to the main diagonal

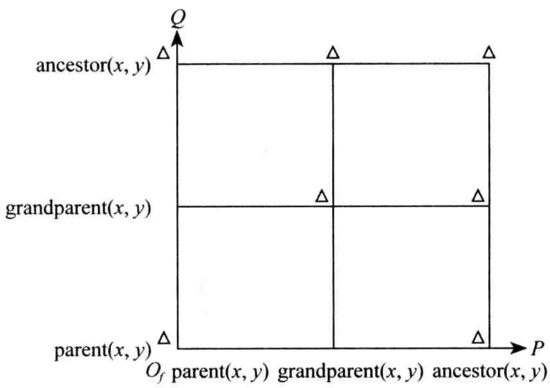


Fig. 9.13 Mutually inverse diagram for compatible connection operator

9.2.6 Quasi-compatible connection operators

Definition 9.23: If an empirical or mathematical connection operator φ on I_l is irreflexive, and symmetric, then φ is called a quasi-compatible connection operator on I_l.

Example 9.22: Suppose $I_l =^{-1} \{\text{publish}(x), \text{perish}(x)\}$, then $\cup|^{-1}$ and $\cup^{-1}$ are quasi-compatible connection operators on I_l, their mutually inverse diagram is shown in Fig. 9.14

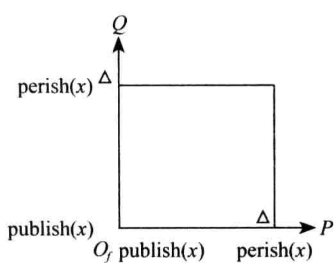


Fig. 9.14 Mutually inverse diagram for $\cup|^{-1}$ and $\cup^{-1}$

Example 9.23: Suppose $\mathrm{I}_1 =^{-1} \{\text{even\_number}(x), \text{odd\_number}(x)\}$, then $\oplus^{-1}$ is a quasi-compatible connection operator on I_1, its mutually inverse diagram is shown in Fig. 9.1.

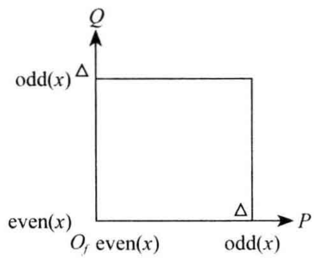


Fig.9.15 Mutually inverse diagram for $\oplus^{-1}$

Example 9.24: Suppose $\mathrm{I}_1 =^{-1} \{\text{rectangle}(x), \text{rhombus}(x)\}$, then $\times^{-1}$ is a quasi-compatible connection operator on I_1.

9.3 The main in mutually-inversistic set theory vs. binary relations in naïve set theory

Relational matrices in naïve set theory are replaced by mutually inverse diagrams in mutually-inversistic set theory. Mutually inverse diagrams can denote transitivity, can be used to study empirical or mathematical connection operators. Mutually-inversistic set theory deems that $|\cap^{-1}$ is a compatible connection operator, and proposes quasi-compatible connection operators. These are the advantages of mutually-inversistic set theory over naïve set theory.

An empty relation is shown in Fig. 9.16, a universal relation is shown in Fig. 9.17. Mutually-inversistic set theory deems that they are meaningless distinguished sets.

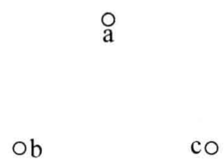


Fig. 9.16 Digraph for empty relation

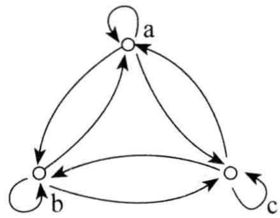


Fig. 9.17 Digraph for universal relation

In mutually-inversistic set theory, antisymmetry is defined as $r(x, y) \cap x \neq y \subseteq^{-1} \sim r(y, x)$ and $\{P\varphi Q\} \cap \{P \neq^{-1} Q\} \subseteq^{-1} \sim \{Q\varphi P\}$, because extended intersections can be performed for them.

Chapter 10 The auxiliary

This chapter studies the second-order auxiliary; i.e., functions, and the third-order auxiliary; i.e., fact composition operators.

10.1 Functions

10.1.1 Unary functions

Suppose X and Y are sets, f is a unary function from X to Y, denoted as f: X→Y, if it satisfies the following conditions:

For every $x \in$X, there exists a unique $y \in$Y, such that $(x, y) \in f$.

$(x, y) \in f$ is usually denoted as $f(x)=y$. X is called the domain of function f, Y the range. In the expression $f(x)=y$, x is called the argument of the function, y the function value corresponding to the argument x.

We only study functions of the form f: X→X, where X is a finite set.

A unary function can be denoted by a mutually inverse diagram on a term plane. For example, the mutually inverse diagram of $y=|x|$ is denoted by the vertices marked with "△" in Fig. 10.1.

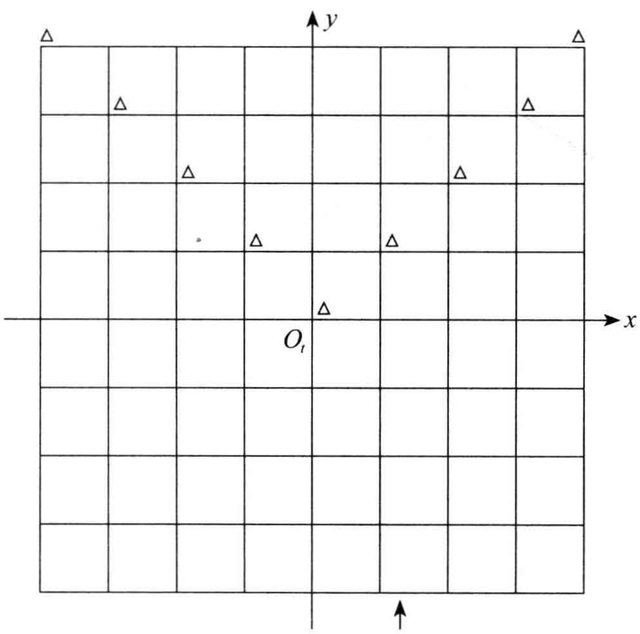


Fig.10.1 Mutually inverse diagram for $y=|x|$

The method for determining whether a mutually inverse diagram on a term plane is a unary function or not is as follows: seeing along the direction of the arrow in Fig. 10.1, if for every x, we see and only see one vertex marked with "△", then the mutually inverse diagram represents a unary function, otherwise, it does not. According to this method, we can decide that the mutually inverse diagram of $y=|x|$ in Fig. 10.1 is a unary function.

Definition 10.1: Suppose f: X→X, where X is a finite set.

(1) If $f(x)$=X, then f is said to be surjective.

(2) If $x \neq x'$ implies $f(x) \neq f(x')$, then f is said to be injective.

(3) If f is both surjective and injective, then f is said to be unary bijective.

Theorem 10.1: Suppose f: X→X, where X is a finite set. F is surjective if and only if f is injective.

Proof: Necessity: Suppose that f is surjective; i.e., x_1 is mapped to x_{i1}, x_2 is mapped to $x_{i2}, \ldots$, x_n is mapped to x_{in}, where $i_1, i_2, \ldots, i_n$ are distinct. Suppose conversely that f is not injective; i.e., both x_j and x_k ($j \neq k$)are mapped to x_{ij}, then x_{ik} is not mapped to. But this contradicts with the supposition that f is surjective, therefore, f is bound to be injective.

Sufficiency: Suppose that f is injective, suppose conversely that f is not surjective; i.e., x_i is not mapped to. Then x_j is mapped to twice; we have $x \neq x'$, and $f(x)=f(x')=x_j$. But this contradicts with the supposition that f is injective, therefore, f is bound to be surjective.

Q.E.D.

From Theorem 10.1, we learn that a unary function is either a unary bijection, or it is neither a surjection nor an injection. Therefore, we need only to consider unary bijection. Unary function $y=-x$ is a unary bijection, shown in Fig. 10.2.

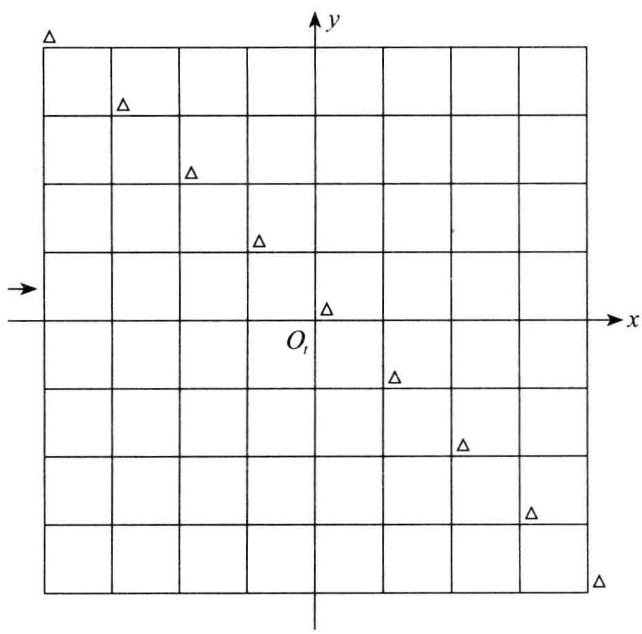


Fig. 10.2 Mutually inverse diagram for unary bijection $y=-x$

Theorem 10.2: Suppose we have a unary function and its mutually inverse diagram. For every y, there is one and only one x in the mutually inverse diagram corresponding to it, if and only if the unary function denoted by the mutually inverse diagram is a unary bijection.

Proof: Sufficiency: Suppose that unary function f is a unary bijection. Suppose conversely that for certain y, there are two x's in the mutually inverse diagram of f corresponding to it. Then f is not an injection, let alone unary bijection, contradicting with the supposition that f is a unary bijection. Suppose conversely that for certain y, there is no x in the mutually inverse diagram of f corresponding to it. Then f is not a surjection, let alone unary bijection, contradicting with the supposition that f is a unary bijection. Therefore, for every y, there is one and only one x in the mutually inverse diagram of f corresponding to it.

Necessity: Suppose we have a unary function f and its mutually inverse diagram, and for every y, there is one and only one x in the mutually inverse diagram corresponding to it. Because for every y, there is one x in the mutually inverse diagram corresponding to it, f is a surjection. Because for every y, there is only one x in the mutually inverse diagram corresponding to it, f is an injection. Since f is both a surjection and an injection, f is a unary bijection.

Q.E.D.

According to Theorem 10.2, the method of determining whether a unary function is a unary bijection or not is as follows: seeing along the direction of the arrow in Fig. 10.2, if for every y, we see one and only one vertex marked with "△", then the unary function is a unary bijection, otherwise, it is not.

According to this method, we can decide that $y=-x$ in Fig. 10.2 is a unary bijection, while $y=|x|$ in Fig. 10.1 is not, because when $y>0$, for each y, we see two vertices marked with "△", when $y<0$, for each y, we see no vertex with "△".

10.1.2 Binary functions

Definition 10.2: The Cartesian product of sets A and B, denoted as A×B, is the set of ordered pairs $\{(a, b)|a\in A\wedge b\in B\}$.

We only study the binary functions from A×A to A, where A is a finite set, denoted as f:A×A→A, satisfying the following condition: for every $(a, b)\in A\times A$, there exists a unique $c\in A$, such that $((a, b), c)\in f$.

A binary function is usually denoted as $z=f(x, y)$. The binary functions in mathematics, such as +, ∗, are usually denoted as $z=x+y$, $z=x*y$.

We can use mutually inverse diagrams in the term space to denote binary functions. For example, the mutually inverse diagram of $z=x+_4y$ ($+_4$ is addition modulo 4) is the vertices marked with "△" in Fig. 10.3.

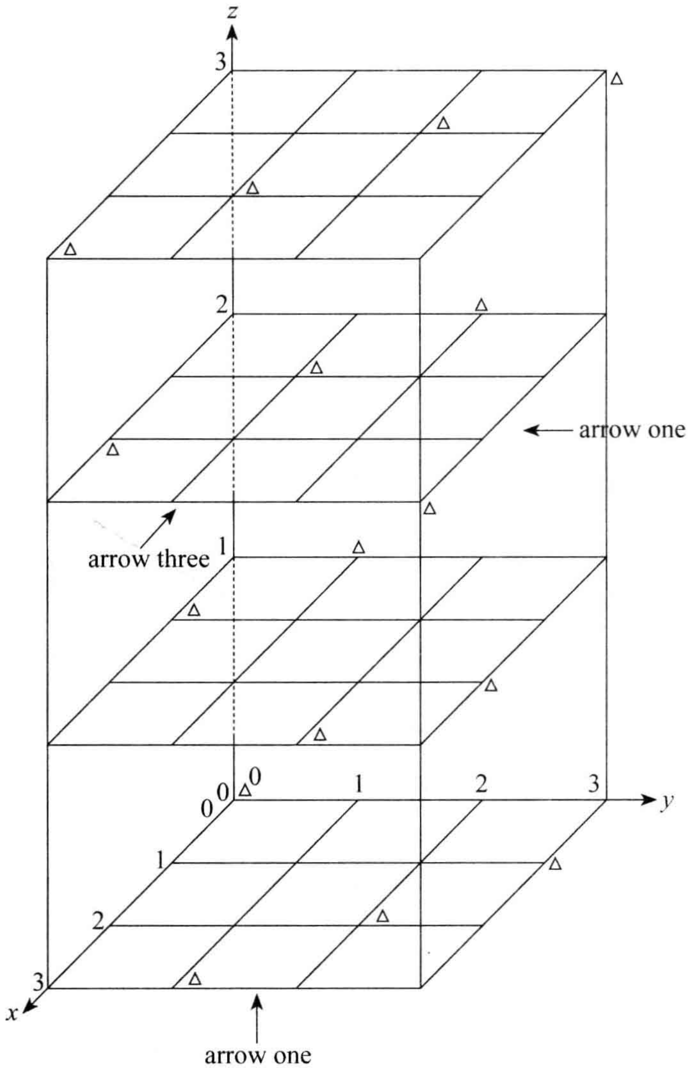


Fig. 10.3 Mutually inverse diagram for $z=x+_4y$

The method for determining whether a mutually inverse diagram in the term space is a binary function or not is as follows: seeing along the direction of arrow one in Fig. 10.3, if for every (x, y) we see one and only one vertex marked with "△", then the mutually inverse diagram represents a binary function, otherwise, it does not.

According to this method, we can decide that the mutually inverse diagram of $z=x+_4y$ in Fig. 10.3 represents a binary function.

Definition 10.3: Suppose f:A×A→A, where A is a finite set, and f is denoted by a mutually inverse diagram in the term space. In the term space, we make sections perpendicular to the y-axis (and the x-axis), if the sections for every vertex of the y-axis (and the x-axis) are unary bijections, then f is a binary bijection.

The mutually inverse diagram of $z=x+_4y$ in Fig. 10.3 is a binary bijection.

Theorem 10.3: Suppose we have a binary function *f* and its mutually inverse diagram. For every (x, z), there is one and only one y in the mutually inverse diagram corresponding to it, and for every (y, z), there is one and only one x in the mutually inverse diagram corresponding to it, if and only if *f* is a binary bijection.

Proof: Sufficiency: Suppose f is a binary bijection. Suppose conversely that for (x_1, z_1) there are two y's: y_1 and y_2 in the mutually inverse diagram corresponding to it. Then we make the section perpendicular to x_1, this section is not an injection (because both y_1 and y_2 are mapped to z_1), let alone a unary bijection. Therefore, *f* is not a binary bijection. But this contradicts with the supposition. We also suppose conversely that for (x_2, z_2) there is no y in the mutually inverse diagram corresponding to it. Then we make the section perpendicular to x_2, this section is not a surjection (because z_2 is not mapped to), let alone a unary bijection. Therefore, *f* is not a binary bijection. But this contradicts with the supposition. Therefore, for every (x, z), there exists one and only one y in the mutually inverse diagram of f corresponding to it. Likewise, for every (y, z), there exists one and only one x in the mutually inverse diagram of *f* corresponding to it.

Necessity: Suppose that for every (x, z), there exists one and only one y in the mutually inverse diagram of *f* corresponding to it, and for every (y, z), there exists one and only one x in the mutually inverse diagram of *f* corresponding to it. Suppose conversely that *f* is not a binary bijection. Do not lose generality, suppose conversely that the section perpendicular to x_1 is not a unary bijection; i.e., both y_1 and y_2 maps to z_1, while z_2 is not mapped to. Then for (x_1, z_1), both y_1 and y_2 correspond to it; for (x_1, z_2), no y corresponds to it. But this contradicts with the supposition. Therefore, *f* is bound to be a binary bijection.

Q.E.D.

According to Theorem 10.3, the method for determining whether a binary function is a binary bijection or not is as follows: first, seeing along the direction of arrow two in Fig. 10.3, if for every (x, z), we see one and only one vertex marked with "△"; then seeing along the direction of arrow three in Fig. 10.3, if for every (y, z), we see one and only one vertex marked with "△"; then the binary function is a binary bijection, otherwise, it is not.

According to this method, we can decide that the mutually inverse diagram for the binary function $z=x+_4y$ in Fig. 10.3 is a binary bijection.

The mutually inverse diagrams of idempotency and complement idempotency needs the concepts of diagonal axes, therefore, we first specify them. In Fig. 10.4, the vertices marked with A constitute the main diagonal axis, the vertices marked with B constitute the left auxiliary diagonal axis, the vertices marked with C constitute the middle auxiliary diagonal axis, the vertices marked with D constitute the right auxiliary diagonal axis.

Definition 10.4: Suppose *f* is a binary function. If for all x, $f(x, x)=x$ holds, then *f* is idempotent.

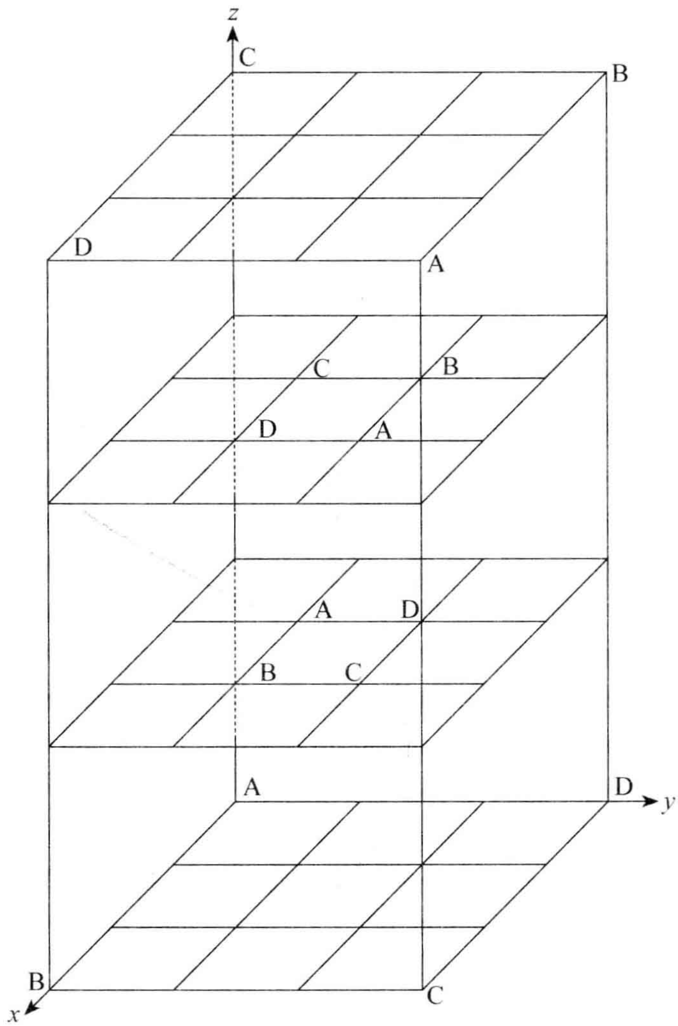


Fig. 10.4 Mutually inverse diagrams for diagonal axes

The pictorial feature of idempotency of *f* is that all the vertices on the main diagonal axis of the term space belong to its mutually inverse diagram. For example, when the universe of terms is the set S_6 of all the factors of 6, S_6={1, 2, 3, 6}, the binary function is GCD (greatest common divisor), its mutually inverse diagram is shown in Fig. 10.5. From Fig. 10.5 we learn that all the vertices on the main diagonal axis: (1, 1, 1), (2, 2, 2), (3, 3, 3), and (6, 6, 6), are in its mutually inverse diagram. Therefore, GCD is idempotent.

A binary function can be both binary bijective and idempotent, see Fig. 10.6.

Definition 10.5: Suppose *f* is a binary function. If for all *x*, we have *f*(*x*, *x*)=*x*' (*x*' denotes the complement of *x*, we will define complement in mutually-inversistic abstract algebra), then *f* is complement idempotent.

For example, when the universe of terms is the set S_6 of all the factors of 6, S_6={1, 2, 3, 6}, the complement operation is stipulated as 1'=6, 2'=3, 3'=2, 6'=1. Suppose the binary

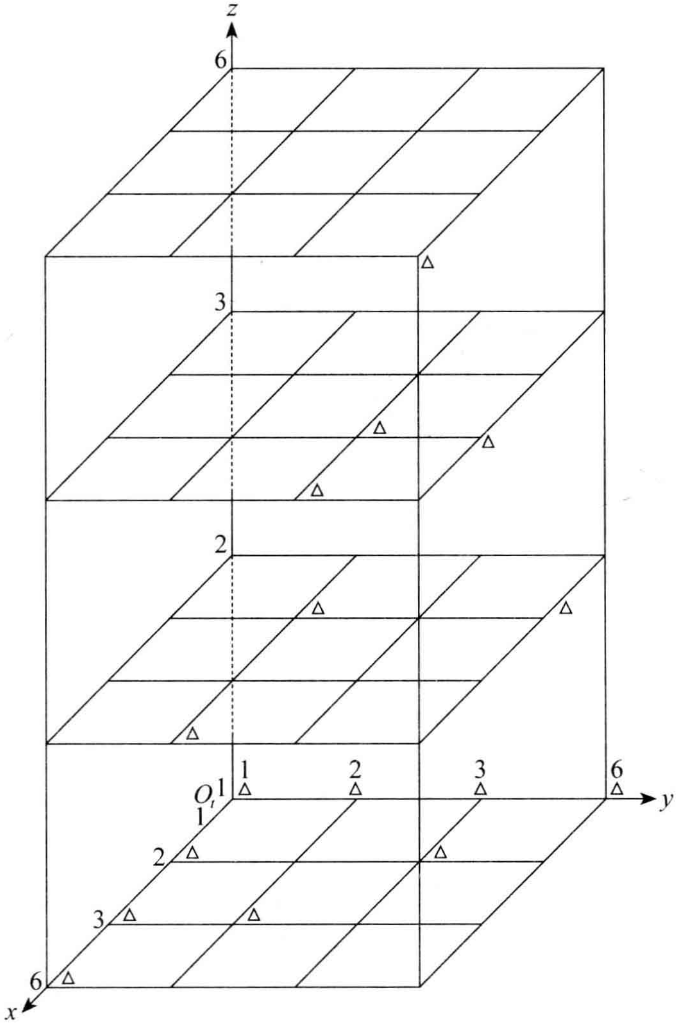


Fig. 10.5 An example of idempotency: z=GCD(x, y)

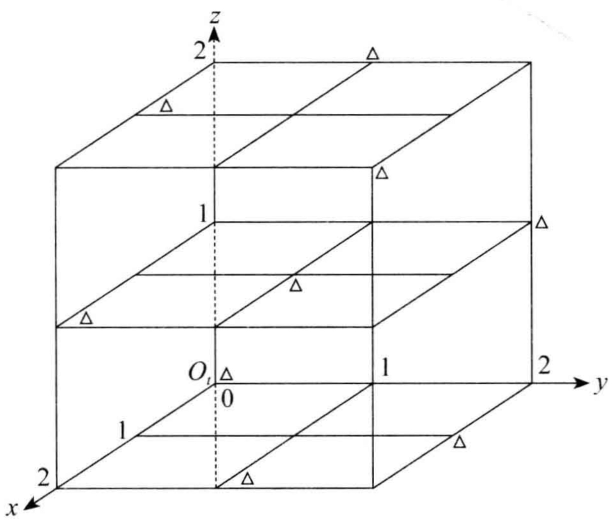


Fig. 10.6 A binary function that is both binary bijective and idempotent

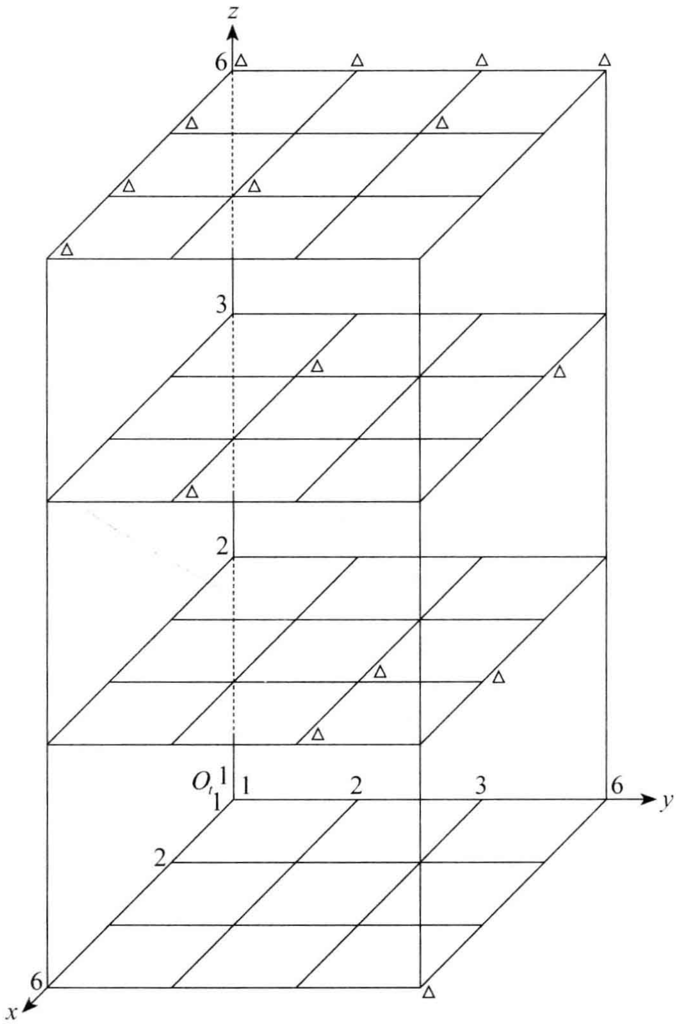


Fig. 10.7 An example of complement idempotency: z=(GCD(x, y))'

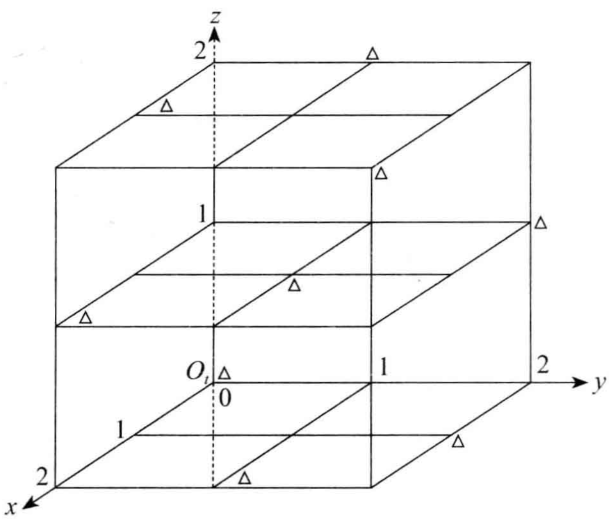


Fig. 10.8 A binary function that is both binary bijective and complement idempotent

function is z=(GCD(x, y))', its mutually inverse diagram is shown in Fig. 10.7. From Fig. 10.7, we learn that all the vertices on the middle auxiliary diagonal axis: (1, 1, 6), (2, 2, 3), (3, 3, 2), and (6, 6, 1), are in its mutually inverse diagram. Therefore, the binary function is complement idempotent.

A binary function can be both binary bijective and complement idempotent, see Fig. 10.8.

Definition 10.6: Suppose f is a binary function. If for all x, y, we have $f(x, y)=f(y, x)$, then f is commutative.

The pictorial feature of a commutative binary function f is that its mutually inverse diagram is symmetric with respect to the main diagonal plane of the term space. For example, the mutually inverse diagram of $z=x+_4y$ (see Fig. 10.3) is symmetric with respect to the main diagonal plane, therefore, $+_4$ is commutative.

10.2 Fact composition operators

10.2.1 A unary fact composition operator

$\sim$ is a unary fact composition operator. The unary bijection theory of the unary fact composition operator is similar to that of the unary functions, so is omitted here. The mutually inverse diagram of $\sim$ is shown in Fig. 10.9, from which we learn that $\sim$ is a binary bijection.

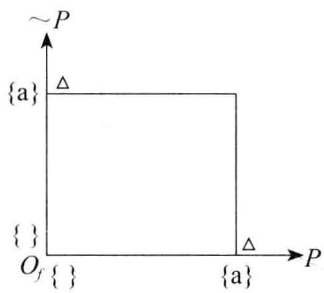


Fig. 10.9 Mutually inverse diagram for $\sim$

10.2.2 Binary fact composition operators

∩, ∪, ⊕, ⊗, ↑(NAND), and ↓(NOR) are binary fact composition operators, their mutually inverse diagrams are shown in Figs. 10.10 to 10.15. The binary bijection, idempotency, complement idempotency, and commutativity theories are similar to those of binary functions, so omitted here. From Fig. 10.10, we learn that ∩ is idempotent and commutative. From Fig. 10.11, we learn that ∪ is idempotent and commutative. From Fig. 10.12, we learn that ⊕ is binary bijective and commutative. From Fig. 10.13, we learn that ⊗ is binary bijective and commutative. From Fig. 10.14, we learn that ↑ is complement idempotent and commutative. From Fig. 10.15, we learn that ↓ is complement idempotent and commutative.

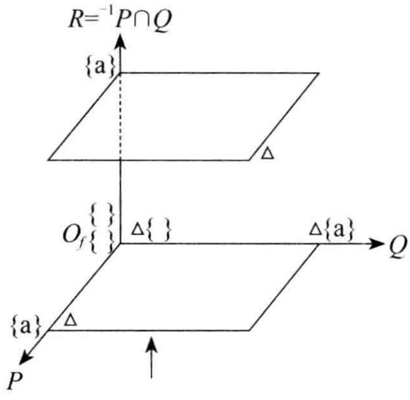


Fig. 10.10 Mutually inverse diagram for ∩

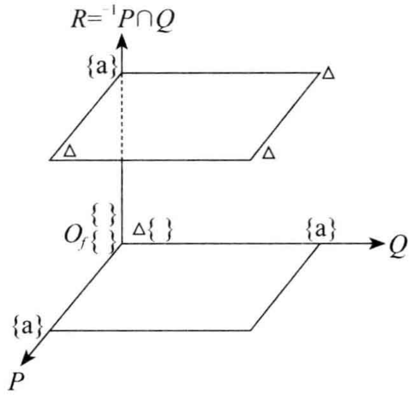


Fig. 10.11 Mutually inverse diagram for ∪

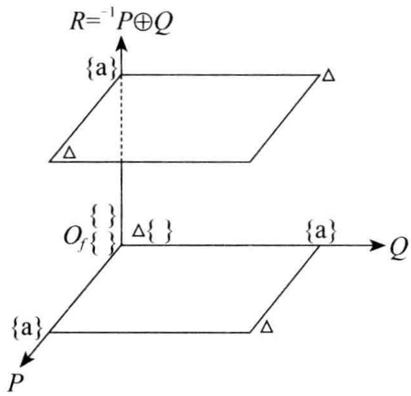


Fig.10.12 Mutually inverse diagram for ⊕

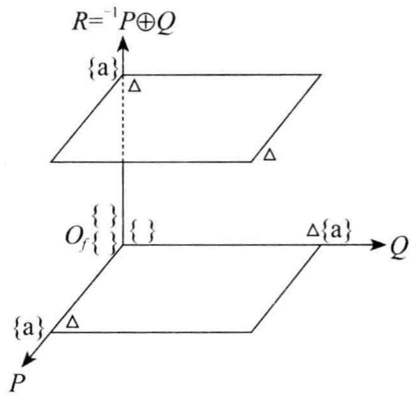


Fig. 10.13 Mutually inverse diagram for ⊗

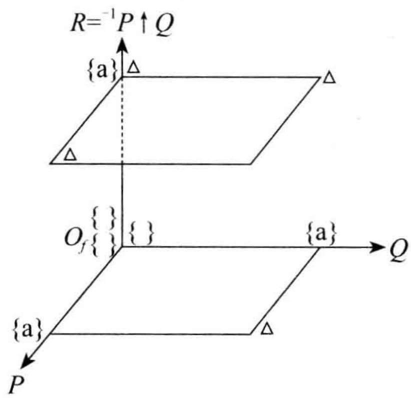


Fig. 10.14 Mutually inverse diagram for ↑

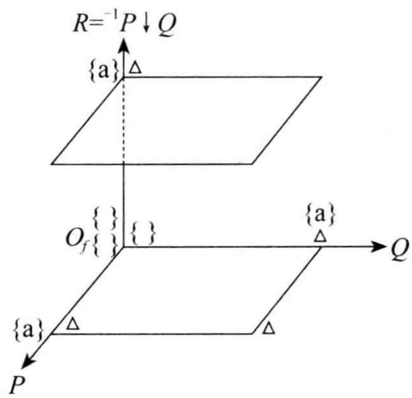


Fig. 10.15 Mutually inverse diagram for ↓

Like binary functions, the method for determining whether a mutually inverse diagram is a binary fact composition operator or not is as follows: seeing along the direction of the arrow in Fig. 10.10, if for every (P, Q), we see one and only one vertex marked with "△", then the mutually inverse diagram is a binary fact composition operator, otherwise, it is not.

Because of this reason, we can crush a binary fact composition operator (and a binary function) along the R-axis, obtaining the two-dimensional operation table. After being crushed, Fig. 10.10 becomes an operation table shown in Table 10.1.

Table 10.1 Operation table for ∩

∩ P \ Q	{ }	{a}
{ }	{ }	{ }
{a}	{ }	{a}

10.3 Reflexivity vs. idempotency, symmetry vs. commutativity

The main has the properties of reflexivity and symmetry. The auxiliary has the properties of idempotency and commutativity. Idempotency can be viewed as three-dimensional reflexivity, commutativity can be viewed as three-dimensional symmetry. In this way, the main can be compared with the auxiliary. For example, $|\cap^{-1}$ satisfies reflexivity and symmetry, while $\cap$ satisfies three-dimensional reflexivity and three-dimensional symmetry. And $\oplus^{-1}$ satisfies irreflexivity and symmetry, while $\oplus$ does not satisfy three-dimensional reflexivity, satisfies three-dimensional symmetry. If mutually inverse diagrams of the auxiliary are crushed into operation tables, then symmetry and commutativity have the same pictorial features.

10.4 The auxiliary in mutually-inversistic set theory vs. functions in naïve set theory

Naïve set theory only studies unary functions. Mutually-inversistic set theory studies not only unary functions but also binary functions, not only functions but also fact composition operators, and studies them with mutually inverse diagrams. In this way, unary bijection, binary bijection, idempotency, complement idempotency, and commutativity can be showed intuitively. These are the advantages of mutually-inversisitc set theory over

naïve set theory.

Classical discrete mathematics studies only unary functions, in order to establish homomorphism and isomorphism between algebras in classical abstract algebra. Mutually-inversistic discrete mathematics studies not only unary functions but also binary functions, not only functions but also fact composition operators, in order to use them as algebraic operators in mutually-inversistic abstract algebra.

10.5 Relations vs. functions

In naïve set theory and axiomatic set theory, a function is a special kind of relation, is single-valued relation. In mutually-inversistic set theory, single-value is not the feature owned only by functions. A relation can also be single-valued; e.g., equality relation=. Relations and functions are different second-order constituents. Relations are the main, while functions are the auxiliary.

Part 3
Mutually-inversistic proof theory vs. mutually-inversistic model theory

This part consists of three chapters. Chapter 11 titled proof theory vs. model theory introduces the origin of proof theory and model theory, the relationship between proof theory and model theory, the circular argument between conventional proof theory and model theory, reveals that the source of the circulation is the model-theoretic semantics of material implication. Chapter 12 titled mutually-inversistic proof theory introduces decomposition axiomatic system, mutually-inversistic elementary number theory, reveals that the proofs of Godel's incompleteness theorems are erroneous. Chapter 13 titled mutually-inversistic model theory introduces model theory of the term space, and model theory of the fact space.

Chapter 11
Proof theory vs. model theory

11.1 Origins of proof theory and model theory

In the probe that whether the parallel postulate in Euclidean geometry is a postulate or a theorem lasting more than two thousand years, two non-Euclidean geometries turned out: Lobachevskian geometry and Riemann geometry. The Beltrami model of the former and the Klein model of the latter marked the start of model theory. The non-contradictions of the non-Euclidean geometries were reduced to those of the Euclidean geometry, Euclidean geometry were reduced to real numbers through Cartesian coordinate system, real numbers to rationals, rationals to natural numbers, natural numbers to sets. Sets belong to logic, and logic is not contradictory. But in 1902, Russell discovered Russell's paradox, causing the third mathematical crisis. In order to overcome paradox, three schools of mathematical logic were proposed. Hilbert proposed Hilbert program that marked the start of proof theory. But Godel's incompleteness theorems declared the infeasibility of Hilbert program.

11.2 Relationship between proof theory and model theory

11.2.1 Axiomatic systems

Starting from axioms, according to certain rules, inferring a series of theorems, this kind of system is called an axiomatic system. Euclidean geometry is an axiomatic system.

11.2.2 Formalized axiomatic systems

A formalized axiomatic system starts from the initial symbols that carry no meaning (interpreted as initial concepts), through formation rules, form well-formed formulas (interpreted as propositions). It chooses some well-formed formulas as the starting point (interpreted as axioms), through transformation rules (interpreted as inference rules), derives derived formulas (interpreted as theorems).

11.2.3 Formal systems

If for a formalized axiomatic system, the following conditions can be mechanically decided (intuitively, it means having definite procedures, deciding in finite steps), then the

system is a formal system:

(1) Whether a symbol is an initial symbol or not?

(2) Whether a symbol sequence is a well-formed formula or not?

(3) Whether a well-formed formula is an initial formula or not?

(4) Whether a formula is derived from given formulas through transformation rules?

11.2.4 Object language vs. metalanguage

Initial symbols and well-formed formulas form the language of a formalized axiomatic system, this language is our object of study, called object language or formal language. When we discuss this system, we also use a language, this language is called a metalanguage, which is usually natural language plus some agreed-on symbols.

11.2.5 Syntactics

A metalanguage is also called a syntactic language. The theory regarding an object language discussed by a syntactic language is called the syntactic theory of the object language, syntactics for short.

11.2.6 Semantics

A formalized axiomatic system can have no interpretations. But generally speaking, a formalized axiomatic system is the formalization of concrete systems. In order to apply it to concrete systems, a formalized axiomatic system needs interpretations. The theory regarding these interpretations is called the semantic theory, semantics for short. A formalized axiomatic system can have different interpretations.

11.2.7 Relationship between syntactics and semantics

A formal language is the common ground of both syntactic derivation and semantic interpretation. The relationship between syntactics and semantics is shown in Fig. 11.1.

formal language
(initial symbols and formation rules)

semantics
(interpretations)

syntactics
(initial formulas and transformation rules)

relationship between syntactics and semantics
(generalized consistency and completeness)

Fig. 11.1 Relationship between syntactics and semantics

Syntactics and semantics are linked by generalized consistency and completeness:

Suppose φ is a sentence (A sentence is a proposition without free variables in it).

Generalized consistency: if $|\!-\varphi$, then $|\!=\varphi$.

Generalized completeness: if $|\!=\varphi$, then $|\!-\varphi$.

Here, $|\!-$ means provable, $|\!=$ means satisfiable.

The relationship between syntactics and semantics is also one between proof theory and model theory.

11.3 Circular argument between conventional proof theory and conventional model theory

There exists circular argument between the declarative semantics and procedural semantics of Prolog. The declarative semantics of Prolog is based on conventional model theory. The procedural semantics of Prolog is based on conventional proof theory. So, the circular argument between the declarative semantics and procedural semantics of Prolog is one between conventional model theory and conventional proof theory.

In order to reveal the circular argument, first, let us do some preliminary work.

Definition 11.1: Suppose L is a first-order language.

The Herbrand universe U_L of L is the set of all the ground terms in L.

The Herbrahd base B_L of L is the set of all the ground atoms in L.

A Herbrand interpretation of L is a subset of B_L.

Example 11.1: Consider a Prolog program:

P_1 (1) q(a, f(a)).

(2) p(X):-q(a, f(X)).

P_1 implicitly defines a first-order language, its U_L and B_L are U_{P1} and B_{P1}.

U_{P1}={a, f(a), f(f(a)), …}.

B_{P1}={p(a), p(f(a)), p(f(f(a))), …

q(a, a), q(a, f(a)), q(a, f(f(a))), …

q(f(a), a), q(f(a), f(a)), q(f(a), f(f(a),)) …}.

Definition 11.2: Suppose S is a set of well-formed formulas of a first-order language L. If a Herbrand interpretation of S is a model of S, then it is called a Herbrand model of S.

If P is a program, then all of its Herbrand interpretations are all of the subsets of B_P (the power set 2^{Bp}), some of the elements in 2^{Bp} are the Herbrand models of P, others are not.

Example 11.2: Consider the Program that defines the "less or equal" relation:

P_2 (1) leq(0, X).

(2) leq(succ(X), succ(Y)):-leq(X, Y).

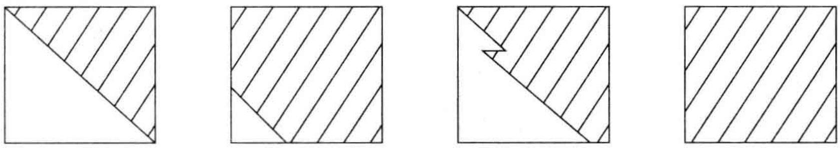
We have

$U_{P2}=\{0, succ(0), succ(succ(0)), \ldots\}$.

$B_{P2}=\{leq(0, 0), leq(0, succ(0)), \ldots$

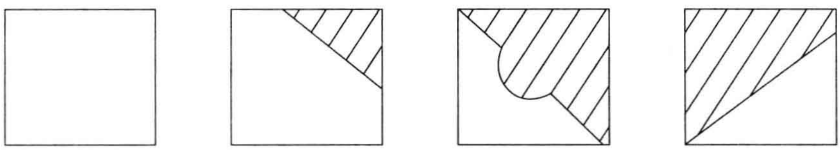
$leq(succ(0), 0), leq(succ(0), succ(0)), \ldots\}$.

B_{P2} can be viewed intuitively as a square. The subsets (denoted by the shadows) of Fig. 11.1 are the Herbrand models of P_2, the subsets of Fig. 11.2 are not the Herbrand models of P_2



(a) (b) (c) (d)

Fig. 11.1 Herbrand models of $_2$



(a) (b) (c) (d)

Fig. 11.2 Non-Herbrand models of P_2

Now, let us reveal the circular argument between the declarative semantics and procedural semantics of Prolog.

Definition 11.3: Suppose S is a set of closed formulas of first-order language L, F is a closed formula of L. If any interpretation I of L being a model of S implies I being a model of F, then F logically follows from S.

Example 11.3: Suppose $S=\{A(e), A(x)\rightarrow B(x)\}$, $F=B(e)$. Then B(e) logically follows from $\{A(e), A(x)\rightarrow B(x)\}$, if and only if all models of $\{A(e), A(x)\rightarrow B(x)\}$ are models of B(e).

Now, let us investigate some of the theorems in reference (Lloyd, 1984). In the theorems that follow, suppose S is a finite set of clauses, and is called a program.

Theorem 11.1 (Theorem 3.2 in (Lloyd, 1984)): Suppose S is a set of clauses and S has a model, then S has a Herbrand model.

Theorem 11.2 (Theorem 6.1 in (Lloyd, 1984)): Suppose S is a Program, $\{M_i\}_{i\in I}$ is a non-empty set of Herbrand models of S. Then $\cap_{i\in I}M_i$ is a Herbrand model of S, denoted by M_p, and is called the least Herbrand model of S.

Theorem 11.3: Suppose S is a program, S has a model if and only if S has M_p.

Proof: Necessity: Theorem 11.1 says, if S has a model, then S a Herbrand model. Theorem

11.2 says, if S has a Herbrand model, then S has M_p. Applying transitive rule to Theorem 11.1 and Theorem 11.2, we obtain that if S has a model, then S has M_p.

Sufficiency: M_p itself is a model.

Theorem 11.3 tells us that the study of model is equivalent to the study of M_p. So, Definition 11.3 is equivalent to Definition 11.4.

Definition 11.4: Suppose S is a set of closed formulas of first-order language L, F is a closed formula of L. If the least Herbrand model M_p of S is also a model of F, then F logically follows from S.

According to Definition 11.4, in Example 11.3, if M_p for $\{A(e), A(x) \rightarrow B(x)\}$ is also a model for B(e), then B(e) logically follows from $\{A(e), A(x) \rightarrow B(x)\}$.

Now, let us construct M_p for $\{A(e), A(x) \rightarrow B(x)\}$.

Theorem 11.4 (Theorem 6.5 in (Lloyd, 1984)): Suppose S is a program, then $M_p = \mathrm{lfp}(T_p) = T_p \uparrow \omega$.

In Theorem 11.4, T_p is a mapping from 2^{Bp} to 2^{Bp}, lfp denotes least fixed point, M_p is the least fixed point of the mapping $T_p: 2^{Bp} \rightarrow 2^{Bp}$: making natural number ω times mapping T_p. T_p is defined as follows:

$T_p(I)=\{D \in B_p | D$ is a ground instance of unconditional clauses in S, or, $D \leftarrow C_1 \wedge C_2 \wedge \ldots \wedge C_n$ is a ground instance of conditional clauses in S, and $\{C_1, C_2, \ldots, C_n\} \in I\}$, where B_p is the Herbrand base, I is the interpretation of L.

Now, first, let us construct the Herbrand universe U_p.

$U_0=\{e\}$.

$U_1=\{e\}=U_0$.

Thus, $U_p=\{e\}$. Then, let us construct $\mathrm{lfp}(T_p)$; i.e., M_p.

$I_0=\varnothing$.

$I_1=T_p(I_0)=\{A(e)\}$.

$I_2=T_p(I_1)=\{A(e), B(e)\}$.

$I_3=T_p(I_2)=\{A(e), B(e)\}=I_2$.

Thus, $M_p=\mathrm{lfp}(T_p)=\{A(e), B(e)\}$.

In Example 11.3, $M_p=\{A(e), B(e)\}$ of $S=\{A(e), A(x) \rightarrow B(x)\}$ includes B(e)=F, that is, M_p is also a model for F, therefore, F logically follows from S.

Theorem 11.5 (Theorem 6.2 in (Lloyd, 1984)): Suppose S is a program, then $M_p = \{D \in B_p |$ D logically follows from S$\}$.

Whether F logically follows from S belongs to the procedural semantics of Prolog. Whether S has M_p belongs to the declarative semantics of Prolog. Definition 11.4 tells us that whether F logically follows from S depend on the construction of M_p for S which is also a model for F. Theorem 11.5 tells us that M_p is composed of those that logically follows from S. Thus, circular argument occurs between the procedural semantics and

the declarative semantics of Prolog: Whether F logically follows from S depends on the construction of M_p for S which is also a model for F, and M_p is composed of all those logically follows from S including F. The procedural semantics of Prolog is based on conventional proof theory. The declarative semantics of Prolog is based on conventional model theory. So, the circular argument is also one between conventional proof theory and conventional model theory.

11.4 Model-theoretic semantics of material implication

Because Table 1.1 has many defects, some people redefine material implication with model-theoretic semantics:

Definition 11.5: Suppose S is a set of closed formulas, F is a closed formula. S implies F, if and only if $S \cup \{\neg F\}$ is unsatisfiable.

Definition 11.5 is equivalent to Definition 11.3, the origin of the circular argument. In order to eliminate the circular argument, material implication cannot be defined by Definition 11.5.

Chapter 12
Mutually-inversistic proof theory

The central idea of proof theory is formalization: mathematical derivation becomes transformation of sequences of artificial symbols, which is free of the ambiguity of natural languages, independent of intuitive properties. After the derivation, people can add meaning to these artificial symbols.

In mutually-inversistic logic, decomposition can be axiomatized and formalized, second-level implicit inductive composition can be algebraized (see Section 23.3 main-auxiliary algebra for set theorems), first-level implicit inductive composition and explicit inductive composition cannot be formalized. Now, we axiomatize the decomposition system of second-level single quasi-predicate calculus.

12.1 Axiomatic system of second-level single quasi-predicate calculus

The knowledge constituents in Section 1.2.2 are the initial concepts. Sections 1.2.4 to 1.2.7 are the formation rules. The well-formed formulas are terms and propositions. In Example 4.5, the known are chosen as the axioms, the five inference rules of decomposition systems are chosen as inference rules, the unknown is the theorem proved.

12.2 Mutually-inversistic elementary number theory

Peano postulates are as follows:

(1) 0 is a natural number;

(2) For every natural number n, there exists another natural number n'(n' is the successor of n);

(3) No natural number n such that $n'=0$;

(4) For any natural numbers m and n, if $m'=n'$, then $m=n$;

(5) For any set A of natural numbers including 0, if for any $n \in A$, we have $n' \in A$, then A contains every natural number.

The formal system of elementary number theory, see (Kleene, 1952), is composed of a set of axiom models of propositional calculus, a set of axiom models of predicate calculus, a set of axioms of number theory. The set of axioms of number theory is as follows:

$A(0)\wedge\forall_x(A(x)\rightarrow A(x'))\rightarrow A(x),$ (12.1)

$x=y\rightarrow x'=y',$ (12.2)

$x'=y'\rightarrow x=y,$ (12.3)

$x=y\rightarrow(x=z\rightarrow y=z),$ (12.4)

$\neg x'=0,$ (12.5)

$x+0=x,$ (12.6)

$x+y'=(x+y)',$ (12.7)

$x*0=0,$ (12.8)

$x*y'=x*y+x.$ (12.9)

In mutually-inversistic logic, formula (12.1) is a cognition. Formulas (12.2) to (12.4) should be written as:

$x=y\leqslant^{-1}x'=y',$ (12.10)

$x'=y'\leqslant^{-1}x=y,$ (12.11)

$x=y\wedge x=z\leqslant^{-1}y=z.$ (12.12)

Formulas (12.10) to (12.12) are single empirical or mathematical connection propositions in number theory, called single number-theoretic propositions, constitute the subsystem of single number-theoretic propositions. Formulas (12.5) to (12.9) are quasi-empirical or mathematical connection propositions in number theory, called quasi-number-theoretic propositions, constitute the subsystem of quasi-number-theoretic propositions.

12.2.1 Subsystem of single number-theoretic propositions

In the subsystem of single number-theoretic propositions, formulas (12.10) to (12.12) are single number-theoretic axioms, the following are single logical axioms:

$\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\},$

$\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\},$

$\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg P\vee/^{-1}Q\},$

$\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\wedge\neg R\leqslant^{-1}\neg Q\},$

$\{P\wedge Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}\neg Q\vee R\}.$

Table 1.3 is the inference rule.

Example 12.1: Prove $\neg\{x'=y'\}\leqslant^{-1}\neg\{x=y\}$.

Proof:

(1) $x=y\leqslant^{-1}x'=y'$ P

(2) $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{\neg Q\leqslant^{-1}\neg P\}$ P

(3) $\neg\{x'=y'\}\leqslant^{-1}\neg\{x=y\}$ T(1)(2)Table1.3

Example 12.2: Prove $x=y\leqslant^{-1}\neg\{x=z\}\vee y=z$.

Proof:

(1) $x=y\wedge x=z\leqslant^{-1}y=z$ P

(2) $\{P \wedge Q \leqslant^{-1} R\} \leqslant^{-1} \{P \leqslant^{-1} \neg Q \vee R\}$ P

(3) $x=y \leqslant^{-1} \neg \{x=z\} \vee y=z$ T(1)(2)Table1.3

12.2.2 Subsystem of quasi-number-theoretic propositions

Quasi-number-theoretic propositions can only be proved by mathematical induction.

Example 12.3: Prove $\neg x'=0$.

Proof: Basis: Let $x=0$, then $0'=1\neq0$.

Induction hypothesis: Suppose when $x=$k, k'=k+1≠0.

Induction step: When $x=$k+1, (k+1)'=k+2≠0.

Q.E.D.

Example 12.4: Prove $x+y'=(x+y)'$.

Proof: Use mathematical induction with two parameters.

Basis: Let $x=0$, $y=0$, then we have 0+0'=0+1=1. (0+0)'=0'=1. Therefore, 0+0'=(0+0)'.

Induction hypothesis: Suppose when $x=$j, $y=$k we have j+k'=(j+k)'.

Induction steps:

(1) When $x=$j+1, $y=$k, the left hand side of the equality is (j+1)+k'=(j+k')+1; the right hand side of the equality is ((j+1)+k)'=((j+k)+1)'=((j+k)')'=(j+k)'+1. Considering the induction hypothesis, the left hand side equals the right hand side; i.e., (j+1)+k'=((j+1)+k)'.

(2) When $x=$j, $y=$k+1, the left hand side of the equality is j+(k+1)'=j+(k')'=j+(k'+1) =(j+k')+1; the right hand side of the equality is (j+(k+1))'=((j+k)+1)'=((j+k)')'= (j+k)'+1. Considering the induction hypothesis, the left hand side equals the right hand side; i.e., j+(k+1)'=(j+(k+1))'.

Therefore, $x+y'=(x+y)'$.

Q.E.D.

12.3 The proofs of Godel's incompleteness theorems are erroneous

This Section reveals that the proofs of Godel's incompleteness theorems are erroneous. This Section refers to the English translation of Godel's original paper (Godel, 1986). The numbers of definitions, formulas, theorems in this Section are the same as those appeared in (Godel, 1986). And the metamathematical notions in this Section are in SMALL CAPITALS.

12.3.1 Godel's first incompleteness theorem

In order to prove his first incompleteness theorem, Godel made many preliminary

works, which are described briefly as follows.

Godel assigned natural numbers to PRIMITIVE SIGNS, finite sequences of PRIMITIVE SIGNS; i.e., FORMULAS, finite sequences of FORMULAS; i.e., PROOF ARRAYS. Later generations call the natural numbers Godel numbers.

The Godel numbers for PRIMITIVE SIGNS are as follows:

"0"...1 "f"(successor function)...3 "$\sim$"...5

"$\vee$"...7 "Π"(universal quantifier)...9 "("...11

")"...13

The Godel numbers for n type VARIABLES are p^n (where p is a prime number greater than 13).

The Godel numbers for FORMULAS are as follows: suppose sequence $n_1, n_2, \ldots, n_k$ is the sequence of Godel numbers of the PRIMITIVE SIGNS occurred in a FORMULA, then $2^{n1} * 3^{n2} * \ldots * p_k^{nk}$ (where p_k is the kth prime number) is the Godel number of the FORMULA.

Definitions (here we only list those that are used in this Section):

6 *nGlx*: the nth term of the number sequence assigned to the number x.

7 $l(x)$: the length of the number sequence assigned to x.

13 Neg(x): the NEGATION of x.

15 xGen$y$: the GENERALIZATION of y with respect to the VARIABLE x.

17 $Z(n)$: the NUMERAL denoting the number n.

31 $Sb(x^v_y)$: VARIABLE v in x is substituted by y.

32 xImp$y$: x implies y.

42 $Ax(x)$: x is an AXIOM.

43 $Fl(x, y, z)$: x is an IMMEDIATE CONSEQUENCE of y and z.

44 $Bw(x)$: x is a PROOF ARRAY.

45 xBy: x is a PROOF of the FORMULA y.

46 Bew(x): x is a PROVABLE FORMULA.

Theorem V: For every recursive relation $R(x_1, \ldots, x_n)$ there exists an n-place RELATION SIGN r (with the FREE VARIABLES $u_1, u_2, \ldots, u_n$) such that for all n-tuples of numbers $(x_1, \ldots, x_n)$ we have

$$R(x_1, \ldots, x_n) \rightarrow \text{Bew}[Sb(r^{u1}_{Z(x1)}{}^{\cdots}_{\cdots}{}^{un}_{Z(xn)}), \tag{3}$$

$$R(x_1, \ldots, x_n) \rightarrow \text{Bew}[\text{Neg}(Sb(r^{u1}_{Z(x1)}{}^{\cdots}_{\cdots}{}^{un}_{Z(xn)}))]. \tag{4}$$

Proof: (Omitted).

Let κ be any class of FORMULAS. We denote by Flg(κ) (the set of consequences of κ) the smallest set of FORMULAS that contains all FORMULAS of κ and all AXIOMS and is closed under the relation "IMMEDIATE CONSEQUENCE". κ is said to be ω-consistent if there is no CLASS SIGN a such that

$(n)[Sb(a^v_{Z(n)}) \in \text{Flg}(\kappa) \& [\text{Neg}(v \text{ Gen } a)] \in \text{Flg}(\kappa),$

Where v is the FREE VARIABLE of the CLASS SIGN a and (n) denotes $\forall n$.

Now, Godel's first incompleteness theorem.

Theorem VI (Godel's first incompleteness theorem): For every ω-consistent recursive class κ of FORMULAS there are recursive CLASS SIGNS r such that neither v Gen r nor Neg(v Gen r) belongs to Flg(κ) (where v is the FREE VARIABLE of r).

Proof: Let κ be any recursive ω-consistent class of FORMULAS. We define

$$Bw_\kappa(x)\equiv(n)[n\leqslant l(x)\rightarrow Ax(nGlx)\vee(nGlx)\in\kappa\vee(Ep,q)\{0<p,q<n\&Fl(nGlx,pGlx,qGlx)\}]\&l(x)>0 \tag{5}$$

(see the analogous notion 44),

$$xB_\kappa y\equiv Bw_\kappa(x)\&[l(x)]Glx=y \tag{6}$$

$$\mathrm{Bew}_\kappa(x)\equiv(Ey)yB_\kappa x \tag{6.1}$$

(see the analogous notions 45 and 46).

We obviously have

$$(x)[\mathrm{Bew}_\kappa(x)\sim x\in \mathrm{Flg}(\kappa)] \tag{7}$$

($\sim$ denotes "equivalence") and

$$(x)[\mathrm{Bew}(x)\rightarrow \mathrm{Bew}_\kappa(x)]. \tag{8}$$

We now define the relation

$$Q(x,y)\equiv\overline{xB_\kappa[Sb(y^{19}_{Z(y)})]}. \tag{8.1}$$

Since $xB_\kappa y$ (by (6) and (5)) and $Sb(y^{19}_{Z(y)})$ (by Definitions 17 and 31) are recursive, so is $Q(x,y)$. Therefore, by Theorem V and (8) there is a RELATION SIGN q (with the FREE VARIABLES 17 and 19) such that

$$\overline{xB_\kappa[Sb(y^{19}_{Z(y)})]}\rightarrow \mathrm{Bew}_\kappa[Sb(q^{17}_{Z(x)}{}^{19}_{Z(y)})] \tag{9}$$

and

$$xB_\kappa[Sb(y^{19}_{Z(y)})]\rightarrow \mathrm{Bew}_\kappa[\mathrm{Neg}(Sb(q^{17}_{Z(x)}{}^{19}_{Z(y)})). \tag{10}$$

We put

$$p=17\ \mathrm{Gen}\ q \tag{11}$$

(p is a CLASS SIGN with the FREE VARIABLE 19) and

$$r=Sb(q^{19}_{Z(p)}) \tag{12}$$

(r is a recursive CLASS SIGN with the FREE VARIABLE 17).

Then we have

$$Sb(p^{19}_{Z(p)})=Sb([17\ \mathrm{Gen}\ q]^{19}_{Z(p)})=17\ \mathrm{Gen}\ Sb(q^{19}_{Z(p)})=17\ \mathrm{Gen}\ r \tag{13}$$

(by (11) and (12)); furthermore

$$Sb(q^{17}_{Z(x)}{}^{19}_{Z(p)})=Sb(r^{17}_{Z(x)}) \tag{14}$$

(by (12)). If we now substitute p for y in (9) and (10), and take (13) and (14) into account, we obtain

$$\overline{xB_\kappa(17\ \mathrm{Gen}\ r)}\rightarrow \mathrm{Bew}_\kappa[Sb(r^{17}_{Z(x)})], \tag{15}$$

$$xB_\kappa(17\ \mathrm{Gen}\ r)\rightarrow \mathrm{Bew}_\kappa[\mathrm{Neg}(Sb(r^{17}_{Z(x)}))]. \tag{16}$$

This yields:

1. 17 Gen r is not κ-PROVABLE. For, if it were, there would (by (6.1)) be an n such that nB_κ(17 Gen r). Hence by (16) we would have

$$\text{Bew}_\kappa[\text{Neg}(Sb(r^{17}_{Z(n)}))],$$

while, on the other hand, from the κ-PROVABILITY of 17 Gen r that of $Sb(r^{17}_{Z(n)})$ follows. Hence, κ would be inconsistent (and a fortiori ω-inconsistent).

2. Neg(17 Gen r) is not κ-PROVABLE. Proof: As has just been proved, 17 Gen r is not κ-PROVABLE; that is (by (6.1)),

$$(n)\overline{nB_\kappa(17 \text{ Gen } r)}$$

holds. From this,

$$(n)\text{Bew}_\kappa[Sb(r^{17}_{Z(n)})]$$

follows by (15), and that, in conjunction with

$$\text{Bew}_\kappa[\text{Neg}(17 \text{ Gen } r)],$$

is incompatible with the ω-consistency of κ.

17 Gen r is therefore undecidable on the basis of κ, which proves Theorem VI.

Q.E.D.

12.3.2 The proof of Godel's first incompleteness theorem is erroneous

Now, let us analyze the errors of the proof of Theorem VI. Because both 17 Gen r and $Sb(r^{17}_{Z(x)})$ are to eliminate the FREE VARIABLE 17 of r, we regard them as identical, as r without the FREE VARIABLE 17, denoted by r'. In doing so, we can simplify our discussion.

In 1, proof by contradiction is used to prove "r' is not κ-PROVABLE". The proof goes as follows: Suppose: "r' is κ-PROVABLE", according to (6.1), we have nB_κ(r'). According to (16), we infer

Bew$_\kappa$[Neg(r')] (the NEGATION of r' is κ-PROVABLE)

But from the κ-PROVABILITY of r' we can also infer r'. Thus, contradiction occurs: Bew$_\kappa$[Neg(r')] and r' (that is, κ is not ω-consistent, contradicting with the supposition that κ is ω-consistent). Therefore, the supposition "r' is κ-PROVABLE" does not hold, its negation "r' is not κ-PROVABLE" is proved.

The error of this proof lies in (16): from "r' is κ-PROVABLE" we cannot infer Bew$_\kappa$[Neg(r')], and contradiction does not occur, the proof by contradiction is invalid. We will analyze the error of (16) later.

In 2, proof by contradiction is also used to prove "Neg(r') is not κ-PROVABLE". Starting from what we have proved in 1 "r' is not κ-PROVABLE" (Bew$_\kappa$[Neg(r')]), by (6.1), we can infer

$(n)\overline{nB_\kappa(r')}$.

From this, by (15), we can infer

$(n)\text{Bew}_\kappa[r']$.

And contradiction occurs: $\text{Bew}_\kappa[\text{Neg}(r')]$ and $(n)\text{Bew}_\kappa[r']$. That is κ is not ω-consistent, contradicting with the supposition that κ is ω-consistent. Therefore, the supposition $\text{Bew}_\kappa[\text{Neg}(r')]$ does not hold, its negation "$\text{Neg}[r']$ is not κ-PROVABLE" is proved.

The error of this proof lies in (15): from "r' is not κ-PROVABLE" we cannot infer (n) $\text{Bew}_\kappa[r']$, and contradiction does not occur, the proof by contradiction is invalid.

Formulas (15) and (16) are wrong. The correct case is: if x is not the κ-PROOF of r', then r' is not κ-PROVABLE. But (15) says: if x is not the κ-PROOF of r', then r' is, on the contrary, κ-PROVABLE. The correct case is: if x is the κ-PROOF of r', then r' is κ-PROVABLE. But (16) says: if x is a κ-PROOF of r', then the negation of r' is κ-PROVABLE.

Now, let us investigate how do these errors occur. Formulas (15) and (16) are obtained form (9) and (10), which are obtained from (8.1) through Theorem V. Formula (3) of Theorem V tells us: from positive R, we can infer the PROVABILITY of positive r. Formula (4) of Theorem V tells us: from negative R, we can infer the PROVABILITY of negative $\text{Neg}(r)$. While when we obtain (9) and (10) from (8.1) and Theorem V, we use Theorem V inversely. Formula (9) is from the negative premise to the positive conclusion. It says: if x is not the κ-PROOF of y, then q is κ-PROVABLE. Formula (10) is from the positive premise to the negative conclusion. It says: if x is the κ-PROOF of y, then q is not κ-PROVABLE. But up to now, we cannot deem that (9) and (10) are wrong, because we have not stipulated the relationship between q and y. If we stipulate that q is identical with $\neg y$, then (9) and (10) are still correct. However, when we use p to substitute for y in (9) and (10), and take into account (13) and (14) to obtain (15) and (16), the error occurs: Using p to substitute for y means that y is identical with p, formula (13) means that p is identical with r', formula (14) means that r' is identical with q; using transitive law twice, we obtain that q is identical with y. While from the above analysis we known that (9) and (10) are correct only if q is identical with $\neg y$. Thus, we start from (9) and(10), through wrong substitutions, obtain the wrong formulas: (15) and (16). Because (15) and (16) are wrong, the proof of Theorem VI is erroneous.

12.3.3 Godel's second incompleteness theorem

Theorem XI (Godel's second incompleteness theorem): Let κ be any recursive consistent class of FORMULAS; then the SENTENTIAL FORMULA stating that κ is consistent is not κ-PROVABLE; in particular, the consistency of P is not provable in P, provided P is consistent (in the opposite case, of course, every proposition is provable [in P]). (P is Principia

mathematica plus Peano axioms).

The proof (briefly outlined) is as follows: Let κ be some recursive class of FORMULAS chosen once and for all for the following discussion (in the simplest case it is the empty class). As appears from 1, page 177 above (1 of the proof of Theorem VI), only the consistency of κ was used in proving that 17 Gen r is not κ-PROVABLE; that is, we have

$$\mathrm{Wid}(\kappa)\rightarrow\overline{\mathrm{Bew}_{\kappa}}(17\ \mathrm{Gen}\ r), \tag{23}$$

that is, by (6.1),

$$\mathrm{Wid}(\kappa)\rightarrow\overline{(x)xB_{\kappa}(17\ \mathrm{Gen}\ r)}.$$

By (13), we have

$$17\ \mathrm{Gen}\ r{=}Sb(p^{19}{}_{Z(p)}),$$

hence

$$\mathrm{Wid}(\kappa)\rightarrow\overline{(x)xB_{\kappa}Sb(p^{19}{}_{Z(p)})},$$

that is, by (8.1),

$$\mathrm{Wid}(\kappa)\rightarrow(x)Q(x,p). \tag{24}$$

We now observe the following: all notions defined (or statements proved) in Section 2, and in Section 4 up to this point, are also expressible (or provable) in P. For throughout we have used only the methods of definition and proof that are customary in classical mathematics, as they are formalized in the system P. In particular, κ (like every recursive class) is definable in P. Let w be the SENTENTIAL FORMULA by which Wid(κ) is expressed in P. According to (8.1), (9), and (10), the relation $Q(x, y)$ is expressed by the RELATION SIGN q, hence $Q(x, p)$ by r (since, by (12), $r{=}Sb(q^{19}{}_{Z(p)})$), and the proposition$(x)Q(x, p)$ by 17 Gen r.

Therefore, by (24), wImp(17 Gen r) is provable in P (and a fortiori κ-PROVABLE).[67] (Footnote 67: That the truth of wImp(17 Gen r) can be inferred from (23) is simply due to the fact that the undecidable proposition 17 Gen r asserts its own unprovability , as was noted at the very beginning.) If now w were κ-PROVABLE, then 17 Gen r would also be κ-PROVABLE, and from this it would follow, by (23), that κ is not consistent.

12.3.4 Godel's second incompleteness theorem is erroneous

The errors in the proof of Godel's second incompleteness theorem:

(1) Formula (23) does not hold, because in proving that 17 Gen r is not κ-PROVABLE, Godel used not only the consistency of κ, but also (16) and (6.1).

(2) Let us retreat one step. Even if (23) holds, that wImp(17 Gen r) is κ-PROVABLE do not hold, what holds is wImp($\overline{\mathrm{Bew}_{\kappa}}$(17 Gen r)). You may think that footnote 67 says that 17 Gen r asserts its own unprovability. Therefore, 17 Gen r is equivalent to $\overline{\mathrm{Bew}_{\kappa}}$(17 Gen r), wImp(17 Gen r) is equivalent to wImp($\overline{\mathrm{Bew}_{\kappa}}$(17 Gen r)).The fact is, first, if Godel were right, 17 Gen r asserts that 17 Gen r is unprovable, then

17 Gen r is a self-reference proposition, hence wImp(17 Gen r)is wrong. Secondly, formula (8.1) does not seem to be a self-referenced proposition. Therefore, wImp($\overline{\text{Bew}_\kappa(17 \text{ Gen } r)}$) is not equivalent to wImp(17 Gen r). Therefore, that wImp(17 Gen r) is κ-PROVABLE does not hold.

(3) Let us retreat one step further. Even if that wImp(17 Gen r) is κ-PROVABLE holds, then from that w is κ-PROVABLE, we cannot infer that 17 Gen r is κ-PROVABLE. Just as from that human beings are animals and that human beings can speak, we cannot infer that animals can speak.

Chapter 13 Mutually-inversistic model theory

Mutually-inversistic model theory includes model theory of term space and model theory of fact space, they are closely related with term space algebra and fact space algebra of mutually-inversistic abstract algebra.

13.1 Model theory of term space

Suppose the formal language of term space $L_t=\{\{p_i\}_{i\in I}, \{f_j\}_{j\in J}, \{c_k\}_{k\in K}\}$, where p_i are predicate constants, f_j are function constants, c_k are term constants, they are all empirical or mathematical symbols; L_t also implicitly includes logical symbols (Note that=is a mathematical symbol, not a logical symbol). The model of L_t is:

$A_t=\langle I_0, \{p_i^A\}_{i\in I}, \{f_j^A\}_{j\in J}, \{c_k^A\}_{k\in K}\rangle$,

where I_0 is the universe of terms, p_i^A are the interpretations of p_i in L_t, f_j^A are the interpretations of f_j, c_k^A are the interpretations of c_k. For the languages commonly used in mathematics, we use the symbols that are customary to denote the universe of terms, predicate constants, function constants, and term constants in models.

Example 13.1: Suppose we have a formal language $L_{t1}=\{\leqslant\}$, then $A_{t1}=\langle \mathbf{N}, \leqslant\rangle$ is its model. Suppose we have a formal language $L_{t2}=\{\leqslant, +, *, 0\}$, then $A_{t2}=\langle \mathbf{R}, \leqslant, +, *, 0\rangle$ is its model.

Sequence $\sigma_t=\langle a_0, a_1, a_2, \ldots\rangle$ formed by choosing some elements from the universe of terms of model A_t is called the assignment of A_t, which uses elements $a_0, a_1, a_2, \ldots$ from I_0 as the interpretations in model A_t of the term variables $x_0, x_1, x_2, \ldots$ in the formal language L_t.

Suppose φ_t is a formula in language L_t, if φ_t is true in model A_t under assignment σ_t, then we say that φ_t is satisfied by σ_t in A_t, denoted as $A_t|=_{\sigma t}\varphi_t$, if φ_t is false, then denoted as $A_t|\neq_{\sigma t}\varphi_t$.

Example 13.2: Suppose $L_t=\{=\}$, $x=y$ is a formula in L_t, its term variables are x and y. $A_t=\langle \mathbf{N}, =\rangle$, $\sigma_{t1}=\langle 1, 1\rangle$, $\sigma_{t2}=\langle 1, 2\rangle$. Then $x=y$ is true in A_t under σ_{t1}, denoted as $A_t|=_{\sigma t1}x=y$; and $x=y$ is false in A_t under σ_{t2}, denoted as $A_t|\neq_{\sigma t2}x=y$.

If φ_t does not contain free term variables, then φ_t is called a sentence. At this time, the truth value of φ_t in model A_t is irrelevant to assignment σ_t. If φ_t is true in A_t under one assignment, then φ_t is necessarily true under other assignments. At this time, we say that φ_t is constantly true in model A_t, denoted as $A_t|=\varphi_t$. If φ_t is constantly true in any model of L_t,

then we say that φ_t is logically true in L_t, denoted as $\models\varphi_t$.

Example 13.3: Suppose $L_t=\{=\}$, $x=x$ is a sentence in L_t. $X=x$ is true for all models of L_t: $A_{t1}=\langle\mathbf{N}, =\rangle$, $A_{t2}=\langle\mathbf{Z}, =\rangle$, $A_{t3}=\langle\mathbf{Q}, =\rangle$, $A_{t4}=\langle\mathbf{R}, =\rangle$, and we have $\models x=x$.

Suppose Γ_t is a set of sentences of L_t, A_t is a model of L_t. If for any sentence $\varphi_t\in\Gamma_t$, we have $A_t\models\varphi_t$, then we say that Γ_t is constantly true in A_t, denoted as $A_t\models\Gamma_t$.

Example 13.4: Suppose $L_t=\{=, +, 0\}$, its model is $A_t=\langle G, =, +, 0\rangle$. If A_t satisfies the following set of sentences:

(1) $(x+y)+z=x+(y+z)$ (associative law)

(2) $x+0=x\wedge 0+x=x$ (0 is the identity of +)

(3) $x\in G\leqslant^{-1} y\in G/\wedge^{-1}\{x+y=0\wedge y+x=0\}$ (there exists inverse element)

then we say that A_t is a group of term space, or sometimes we say that G is a group of term space. If, in addition, A_t satisfies:

(4) $x+y=y+x$ (commutative law)

then we say that A_t is an Abelian group of term space.

13.2 Model theory of fact space

Suppose we have a formal language of fact space $L_f=\{\{|\cap^{-1}, \cup|^{-1}, \subseteq^{-1}, \ldots\}, \{\cup, \cap, \sim, \ldots\}, \{\varnothing, U\}\}$. The model of L_f is $A_f=\langle I_1, \{|\cap^{-1}, \cup|^{-1}, \subseteq^{-1}, \ldots\}, \{\cup, \cap, \sim, \ldots\}, \{\varnothing, U\}\rangle$.

Sequence $\sigma_f=\langle\varnothing, \{a\}, \{b\}, \{a, b\}, \ldots\rangle$ formed by choosing some elements from universe of facts I_1 of model A_f is called the assignment of A_f, which uses elements $\varnothing$, $\{a\}$, $\{b\}$, $\{a, b\}$, ... in I_1 as the interpretations in model A_f of the fact proposition variables P, Q, and R in formal language L_f.

Suppose φ_f is a formula of language L_f. If φ_f is true in model A_f under the assignment σ_f, then we say that φ_f is satisfied by σ_f in A_f, denoted as $A_f|=_{\sigma f}\varphi_f$. If φ_f is false, then denoted as $A_f|\neq_{\sigma f}\varphi_f$.

Example 13.5: Suppose $L_f=\{=^{-1}, S\}$, where $S=\{a, b\}$. $P=^{-1}Q$ is a formula of L_f, its fact proposition variables are P and Q. $A_f=\langle\rho(S), =^{-1}, S\rangle$, $\sigma_{f1}=\langle\{a\}, \{a\}\rangle$, $\sigma_{f2}=\langle\{a\}, \{a, b\}\rangle$. Then $P=^{-1}Q$ is true in A_f under σ_{f1}, we have $A_f|=_{\sigma f1}P=^{-1}Q$. $P=^{-1}Q$ is false in A_f under σ_{f2}, we have $A_f|\neq_{\sigma f2}P=^{-1}Q$.

If φ_f does not contain free fact proposition variables, then φ_f is called a sentence. At this time, the truth value of φ_f in model A_f is irrelevant to assignment σ_f. If φ_f is true in A_f under one assignment, then φ_f is necessarily true under other assignments. At this time, we say that φ_f is constantly true in model A_f, denoted as $A_f\models\varphi_f$. If φ_f is constantly true in any model of L_f, then we say that φ_f is logically true in L_f, denoted as $\models\varphi_f$.

Example 13.6: Suppose $L_f=\{=^{-1}\}$. $P=^{-1}P$ is a sentence in L_f. $P=^{-1}P$ is true for model $A_f=\langle I_1, =^{-1}\rangle$ in L_f, we have $A_f\models P=^{-1}P$.

Suppose Γ_f is a set of sentences of L_f, A_f is a model of L_f. If for any sentence $\varphi_f \in \Gamma_f$, we have $A_f \models \varphi_f$, then we say that Γ_f is constantly true in A_f, denoted as $A_f \models \Gamma_f$.

Example 13.7: Suppose $L_f = \{=^{-1}, \cup, \cap, \sim, \varnothing, S\}$, where S is a non-empty set. $A_f = \langle \rho(S), =^{-1}, \cup, \cap, \sim, \varnothing, S \rangle$. If A_f satisfies the following set of sentences:

(1) $P\cup\{Q\cup R\}=^{-1}\{P\cup Q\}\cup R$, $P\cap\{Q\cap R\}=^{-1}\{P\cap Q\}\cap R$ (associative laws)

(2) $P\cup Q=^{-1}Q\cup P$, $P\cap Q=^{-1}Q\cap P$ (commutative laws)

(3) $P\cup P=^{-1}P$, $P\cap P=^{-1}P$ (idempotent laws)

(4) $P\cup\{P\cap Q\}=^{-1}P$, $P\cap\{P\cup Q\}=^{-1}P$ (absorption laws)

then we say that A_f is a lattice of fact space. If, in addition, A_f satisfies:

(5) $P\cup Q\cap R=^{-1}\{P\cup Q\}\cap\{P\cup R\}$, $P\cap\{Q\cup R\}=^{-1}P\cap Q\cup P\cap R$ (distributive laws)

then we say that A_f is a distributive lattice of fact space. If, in addition, A_f satisfies:

(6) $\sim\{P\cup Q\}=^{-1}\sim P\cap\sim Q$, $\sim\{P\cap Q\}=^{-1}\sim P\cup\sim Q$ (quasi-De Morgan's laws)

(7) $\varnothing\neq^{-1}S$, $P\cup\varnothing=^{-1}P$, $P\cup S=^{-1}S$, $P\cap\varnothing=^{-1}\varnothing$, $P\cap S=^{-1}P$, $\sim S=^{-1}\varnothing$, $\sim\varnothing=^{-1}S$

(zero-one laws)

(8) $P\cup\sim P=^{-1}S$ (law of excluded middle)

$P\cap\sim P=^{-1}\varnothing$ (non-contradiction law)

$\sim\sim P=^{-1}P$ (double complement law)

then we say that A_f is a Boolean algebra of fact space.

Part 4
Mutually-inversistic recursion theory

Mutually-inversistic recursion theory inherits classical recursion theory as its first-level recursion theory, and proposes second-level recursion theory. Mutually-inversistic recursion theory is described from the logic programming perspective.

Chapter 14 Mutually-inversistic recursion theory

14.1 Prolog

Example 14.1: The known: Bob is Max's ancestor, Max is Sam's ancestor, if x is y's ancestor and y is z's ancestor, then x is z's ancestor. The unknown: Bob is Sam's ancestor: The Prolog program and goal of this example are shown as follows:

(1) ancestor(Bob, Max).

(2) ancestor(Max, Sam).

(3) ancestor(x, z): -ancestor(x, y), ancestor(y, z).

(4) ?-ancestor(Bob, Sam).

Clauses (1) and (2) are unconditional clauses. Clause (3) is a conditional clause, in which ancestor(x, z) is the head, ancestor(x, y) and ancestor(y, z) are the body of the conditional clause. The predicate ancestor occurs in both the head and the body, it is a recursive predicate, the recursion is called first-level recursion. An unconditional clause and a conditional clause are called by a joint name program clause. Clause (4) is a goal clause, in which ?-means $\neg$. Prolog can be regarded as based on mutually-inversistic first-level single quasi-predicate calculus: the unconditional clauses are zeroth-order fact propositions, the conditional clause is a first-order single empirical or mathematical connection proposition, the goal clause is a fact proposition.

The process from the program to the goal is a kind of proof by contradiction, called SLD resolution refutation, which can be described by searching the SLD tree. The search strategy is: from left to right, top-down, depth-first plus backtracking. "from left to right" means the leftmost subgoal are always selected as a resolver to participate in the resolution. The other resolver is a program clause. "top-down" means that if the leftmost subgoal can resolve with more than one program clauses (candidate clauses), then in the top-down order.

The SLD tree of Example 14.1 is shown in Fig. 14.1.

In Fig. 14.1, every downward line represents a resolution. On the top of the line is the leftmost subgoal, one resolver. On the left of the line is the number of a program clause, the other resolver. On the right of the line is the substitution made between the two resolvers. On the bottom of the line is the resolvent.

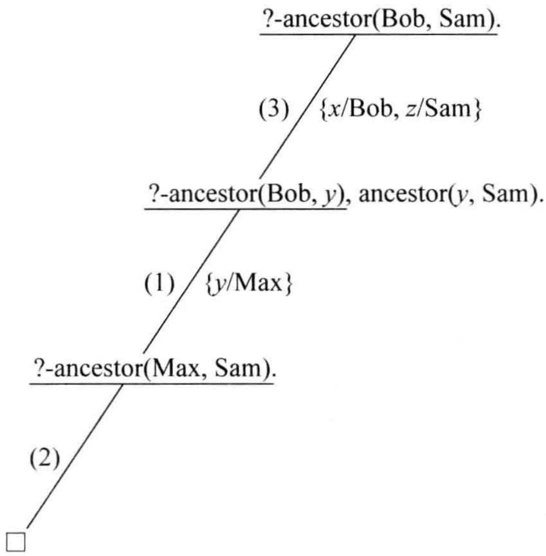


Fig. 14.1 SLD tree for Example 14.1

The first resolution (the first downward line) goes as follows: when x is substituted by Bob, z by Sam (on the right of the line), the initial goal ?-ancestor(Bob, Sam) (on the top of the line) unifies successfully with the head ancestor(x, z) of program clause (3) (on the left of the line), and the body ancestor(Bob, y), ancestor(y, Sam) of (3) is produced (on the bottom of the line). What this resolution actually does is the negative expression of hypothetical inference: from the negation of the conclusion $\neg$ancestor(Bob, Sam) and the major premise ancestor(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z) to infer the negation of the minor premise $\neg${ancestor(Bob, y)$\wedge$ancestor(y, Sam)}, which is equivalent to $\neg$ancestor(Bob, y)$\vee$ $\neg$ancestor(y, Sam), which can be changed to $\neg$ancestor(Bob, y)$\vee/^{-1}\neg$ancestor(y, Sam).

What the second resolution (the second downward line) does is disjunctive inference: from $\neg$ancestor(Bob, y)$\vee/^{-1}\neg$ancestor(y, Sam) and ancestor(Bob, Max) to infer $\neg$ancestor(Max, Sam). What the third resolution (the third downward line) does is a special kind of disjunctive inference: from $\neg$ancestor(Max, Sam) and ancestor(Max, Sam) to produce an empty clause (NIL), which means contradiction occurs, which also ends a success branch.

Example 14.2: Suppose we have the following Prolog program and goal:

(1) parent(Bob, Max).

(2) parent(Max, Sam).

(3) ancestor(x, z): -parent(x, z).

(4) ancestor(x, z): -parent(x, y), ancestor(y, z).

(5) ?-ancestor(Bob, Sam).

In (4), the first predicate in the body is not a recursive predicate, but the last predicate

is. This recursion is called tail recursion. Clause (3) is the recursion exit.

A first-level tail recursion with exit can be transformed into iteration.

Program transformation:

Input: if $f(x, z)$=f(a, c) then h(a, c)

else $h(x, y)$, $f(y, z)$

Output: read(x, z)

L: if $f(x, z)$=f(a, c) then h(a, c)

else $h(x, y)$, $x \leftarrow y$

goto L fi

According to the program transformation, Example 14.2 can be transformed into iteration:

Read ancestor(x, z),

L:if ancestor(x, z)=ancestor(Max, Sam) then parent(Max, Sam)

else parent(x, y), $x \leftarrow y$

goto L fi

(In the iteration, $x \leftarrow y$ means the value of y is assigned to x).

14.2 Second-level single quasi-Prolog

Second-level single quasi-Prolog is obtained by lifting Prolog up one level; i.e., the unconditional clauses are lifted from zeroth-order fact propositions to first-order single empirical or mathematical connection propositions, the conditional clauses are lifted from first-order single empirical or mathematical connection propositions to second-order single logical connection propositions, the goal clauses are lifted from fact propositions to single empirical or mathematical connection propositions. Prolog is based on first-level single quasi-predicate calculus. Second-level single quasi-Prolog is based on second-level single quasi-predicate calculus.

In second-level single quasi-Prolog, the logical connection operator $\leqslant^{-1}$ is denoted by <<<===, the empirical or mathematical connection operators $/\wedge^{-1}$, $\leqslant^{-1}$, $=^{-1}$, and $<^{-1}$ are denoted by!!, <<==, ==, and << respectively. The predicates used in this section are real (real numbers), rat (rationals), int (integers), nat (natural numbers), nonnega (non-negative integers), pos_zero (positive integers and zero), pos_odd (positive odd integers).

Example 14.3: The known: rat(x)$\leqslant^{-1}$real(x), int(x)$\leqslant^{-1}$rat(x), $\{P\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$, $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$. The unknown: int($x$)$/\wedge^{-1}$real($x$).

The second-level single quasi-Prolog program and goal are as follows.

(1) real(x) ==>> rat(x).

(2) rat(x) ==>> int(x).

 (3) {R!!P} ===>>>{R==>>P}.

(4) {R!!P}===>>>{R==>>Q}, {Q!!P}.

(5) ?-{real(x) !! int(x)}.

In (4), the empirical or mathematical connection operator!! occurs in both the head and the body, this is second-level recursion.

In Prolog, there are many predicates. We use recursive predicates only if we want to adopt recursive technique. While in second-level single quasi-Prolog, there are only four empirical or mathematical connection operators, second-level recursion is unavoidable.

14.2.1 One second-order single empirical or mathematical connection proposition in the body

Example 14.4: The known: nonnega(x)$=^{-1}$nat(x), {$P=^{-1}Q$}$\leqslant^{-1}${$Q=^{-1}P$}. The unknown: nat(x)$=^{-1}$nonnega(x).

The second-level single quasi-Prolog program and goal are as follows.

(1) nat(x)==nonnega(x).

(2) {P==Q}===>>>{Q==P}.

(3) ?-{nonnega(x)==nat(x)}.

In (2), there is only one second-order single empirical or mathematical connection proposition in the body: {Q==P}; ==occurs in both the head and the body, this is second-level recursion. The second-level SLD tree of the program and the goal are shown in Fig. 14.2.

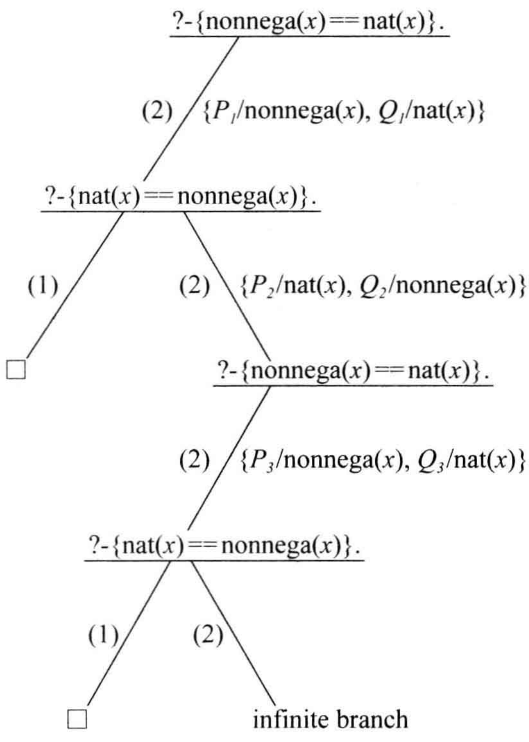


Fig. 14.2 One single empirical or mathematical connection proposition in the body

From Fig. 14.2 we learn that there are infinitely many success branches in its second-level SLD tree.

14.2.2 Two second-order single empirical or mathematical connection propositions in the body

Second-level recursions with two second-order single empirical or mathematical connection propositions in the body are divided into second-level tail recursions and second-level non-tail recursions. In (4) of Example 14.3, the first empirical or mathematical connection operator in the body is not a recursive one, the last empirical or mathematical connection operator in the body is. It is second-level tail recursion.

$$\{R{=}{=}{>}{>}P\}{=}{=}{=}{>}{>}{>}\{R{=}{=}{>}{>}Q\},\ \{Q{=}{=}{>}{>}P\} \tag{14.1}$$

In (14.1), the first empirical or mathematical connection operator in the body is a recursive one. It is second-level non-tail recursion.

14.2.2.1 Second-level tail recursion

Second-level tail recursions are divided into second-level tail recursions with exit and second-level tail recursions without exit.

14.2.2.1.1 Second-level tail recursion with exit

Example 14.5: The known: int(x)$<^{-1}$rat(x), rat(x)$<^{-1}$real(x), $\{P<^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$, $\{P<^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$. The unknown: int($x$)$\leqslant^{-1}$real($x$).

The second-level single quasi-Prolog program and goal are as follows.

(1) rat(x)>>int(x).

(2) real(x)>>rat(x).

(3) $\{R{=}{=}{>}{>}P\}{=}{=}{=}{>}{>}{>}\{R{>}{>}P\}$.

(4) $\{R{=}{=}{>}{>}P\}{=}{=}{=}{>}{>}{>}\{Q{>}{>}P\},\ \{R{=}{=}{>}{>}Q\}$.

(5) ?-{real(x)==>>int(x)}.

Clause (4) is a second-level tail recursion, clause (3) is the recursion exit. The second-level SLD tree of the program and the goal is shown in Fig. 14.3.

From Fig. 14.3, we see that there is a failure branch and a success branch.

Second-level tail recursion with exit can be transformed into second-level iteration by second-level program transformation.

Second-level program transformation:

Input: if $P\psi R\equiv p_2(x)\psi p_3(x)$ then $P\omega R$
else $\{P\theta Q\}\eta\{Q\psi R\}$

Output: read($P\psi R$)

L: if $P\psi R\equiv p_2(x)\psi p_3(x)$ then $P\omega R$
else $P\theta Q$, $P\leftarrow Q$

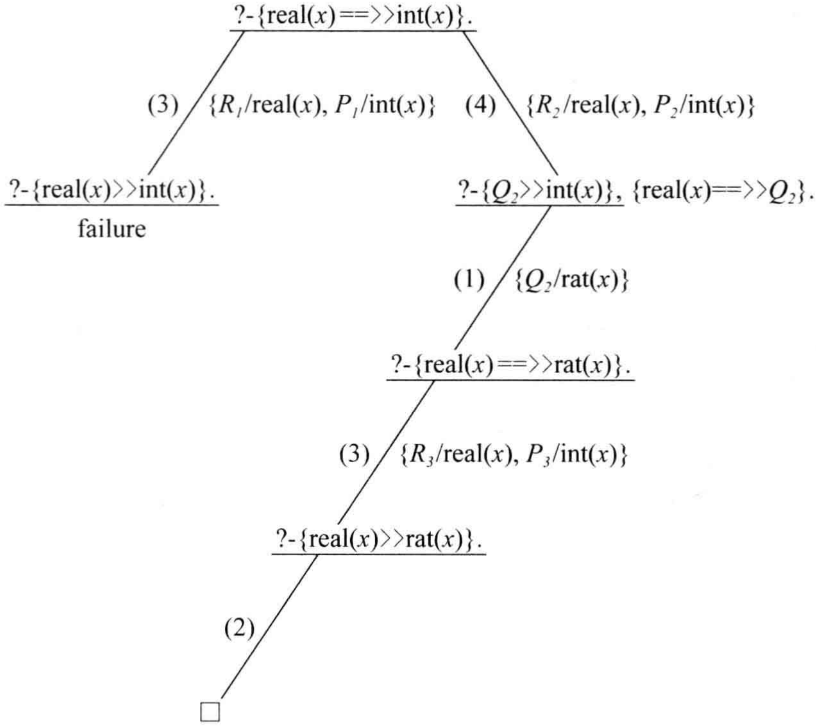


Fig. 14.3 Second-level tail recursion with exit

goto L fi

After the transformation, Example 14.5 becomes the following second-level iteration:

Read(R==>>P)

L: if {R==>>P}≡{real(x)==>>rat(x)} then {real(x)>>rat(x)}
else {Q>>P}, $P \leftarrow Q$

goto L fi

The second-level iteration is performed as follows:

Step 1: Read(R==>>P) statement reads {real(x)==>>int(x)};

Step 2: L statement compares the value of {R==>>P} with {real(x)>>rat(x)};

Step 3: They do not equal, control transfers to {Q>>P};

Step 4: {Q>>P} unifies with (1) {rat(x)>>int(x)} successfully, Q binds to rat(x);

Step 5: $P \leftarrow Q$ statement assign the value of Q; i.e., rat(x), to P;

Step 6: Control transfers to L statement again;

Step 7: {R==>>P} equals {real(x)==>>rat(x)}, control transfers to {R>>P};

Step 8: {R>>P} unifies with (2) {real(x)>>rat(x)} successfully;

Step 9: Program terminates.

14.2.2.1.2 Second-level tail recursion without exit

Example 14.6: The known: nonnega(x)$=^{-1}$nat(x), pos_odd(x)$<^{-1}$nonnega(x), {P

$<^{-1}Q\}\wedge\{Q=^{-1}R\}\leqslant^{-1}\{P<^{-1}R\}$. The unknown: pos_odd($x$)$<^{-1}$nat($x$).

The second-level program and goal are as follows:

(1) nat(x)==nonnega(x).

(2) nonnega(x)>>pos_odd(x).

(3) {R>>P}===>>>{R==Q}, {Q>>P}.

(4) ?-{nat(x)>>pos_odd(x)}.

Clause (3) is a second-level tail recursion. There is no recursion exit. The second-level SLD tree is shown in Fig. 14.4.

?-{nat(x)>>pos_odd(x)}.

(3) {R/nat(x), P/pos_odd(x)}

?-{nat(x)==Q}, {Q>>pos_odd(x)}.

(1) {Q/nonnega(x)}

?-{nonnega(x)>>pos_odd(x)}.

(2)

□

Fig. 14.4 Second-level tail recursion without exit

From Fig. 14.4, we see that there is no infinite branch.

14.2.2.2 Second-level non-tail recursion

In second-level non-tail recursion, infinite branch is bound to occur. When there is success branch, two cases exist: one is that the success branch occurs prior to the infinite branch, the other is that the infinite branch occurs prior to the success branch.

14.2.2.2.1 Success branch prior to infinite branch

Example 14.7: The known: rat(x)$\leqslant^{-1}$real(x), int(x)$\leqslant^{-1}$rat(x), $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$. The unknown: int($x$)$\leqslant^{-1}$real($x$).

The second-level single quasi-Prolog program and goal are as follows.

(1) real(x)==>>rat(x).

(2) rat(x)==>>int(x).

(3) {R==>>P}===>>>{R==>>Q},{Q==>>P}.

(4) ?-{real(x)==>>int(x)}.

The second-level SLD tree is shown in Fig. 14.5.

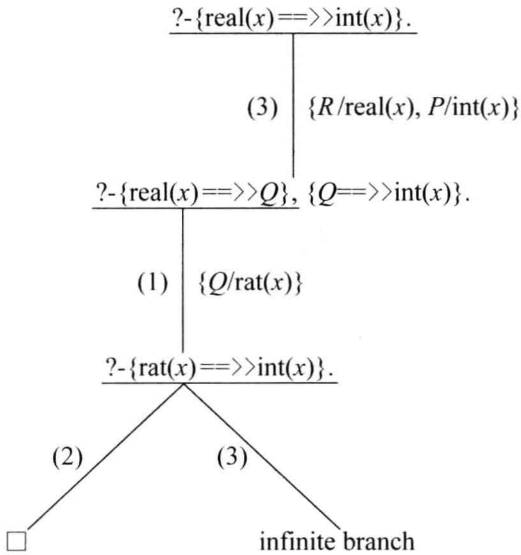


Fig. 14.5 Success branch prior to infinite branch

From Fig. 14.5. we see that success branch is prior to infinite branch.

14.2.2.2.2 Infinite branch prior to success branch

Example 14.8: The known: nonnega(x)$=^{-1}$nat(x), nonnega(x)$=^{-1}$pos_zero(x), $\{P=^{-1}Q\}\wedge\{Q=^{-1}R\}\leqslant^{-1}\{P=^{-1}R\}$, $\{P=^{-1}Q\}\leqslant^{-1}\{Q=^{-1}P\}$. The unknown: pos_zero($x$)$=^{-1}$nat($x$).

The second-level single quasi-Prolog program and goal are as follows.

(1) nat(x)==nonnega(x).

(2) pos_zero(x)==nonnega(x).

(3) {R==P}===>>>{R==Q}, {Q==P}.

(4) {P==Q}===>>>{Q==P}.

(5) ?-{nat(x)==vpos_zero(x)}.

The second-level SLD tree is shown in Fig. 14.6. From Fig. 14.6, we see that the infinite branch comes first. If we do not take measures, we cannot reach the success branch. The measure we take is that once the infinite branch is detected, it is cut, and backtracking; in this way, the success branch can be reached.

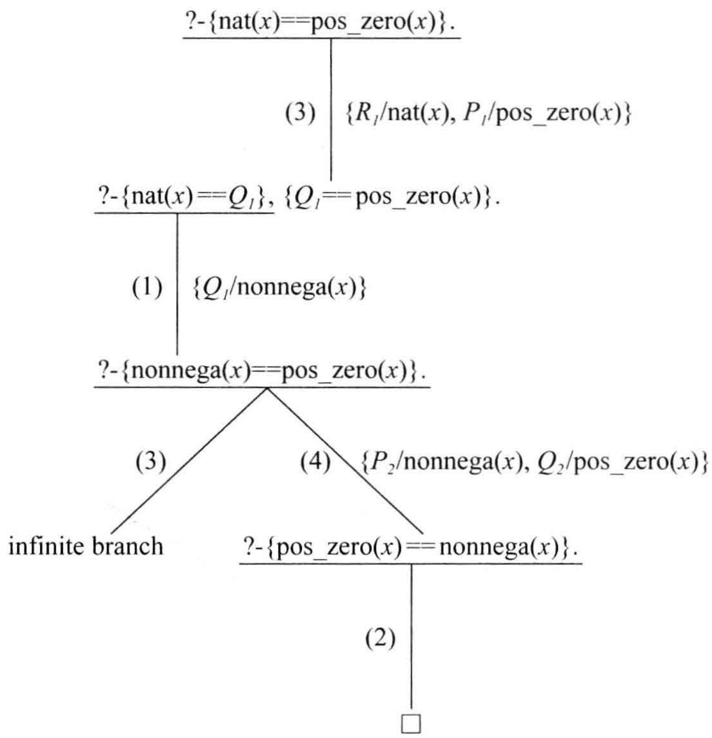


Fig. 14.6 Infinite branch prior to success branch

14.2.3 Three second-order single empirical or mathematical connection propositions in the body

Dilemmas, including constructive dilemma and destructive dilemma, are the second-level recursions with three second-order single empirical or mathematical connection propositions in the body. They are second-level tail recursions without exit.

Example 14.9: The known: prime(x)$/\wedge^{-1}$pos_even(x) (some prime numbers are positive even numbers), prime(x)$\leqslant^{-1}$pos_int(x) (all prime numbers are positive integers), pos_even(x)$\leqslant^{-1}$even(x) (all positive even numbers are even numbers), $\{P\leqslant^{-1}Q\}\wedge\{R\leqslant^{-1}S\}\wedge\{P/\wedge^{-1}R\}\leqslant^{-1}\{Q/\wedge^{-1}S\}$. The unknown: pos_int($x$)$/\wedge^{-1}$even($x$).

The second-level single quasi-Prolog program and goal are as follows.

(1) prime(x)!! pos_even(x).

(2) pos_int(x)==>>prime(x).

(3) even(x)==>>pos_even(x).

(4) {Q!!S}===>>>{Q==>>P}, {S==>>R}, {P!!R}.

(5) ?-{pos_int(x)!!even(x)}.

The second-level SLD tree is shown in Fig. 14.7.

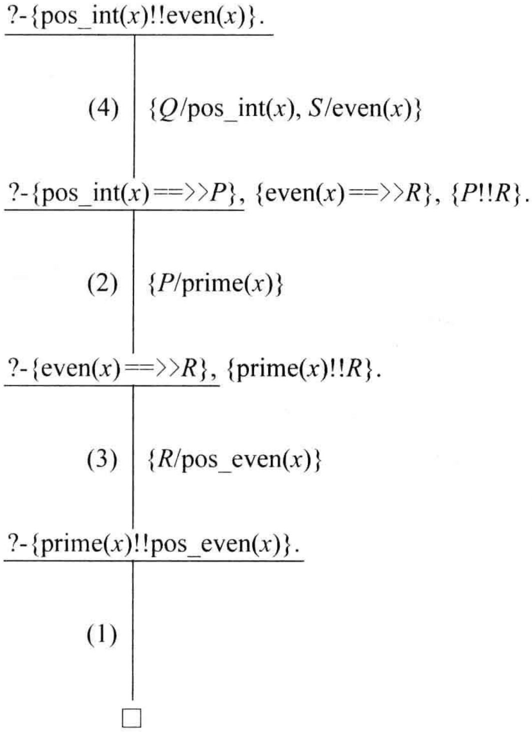


Fig.14.7 Three single empirical or mathematical connection propositions in the body

14.2.4 Summary of second-level recursion

The classification of second-level recursions is shown in Fig. 14.8.

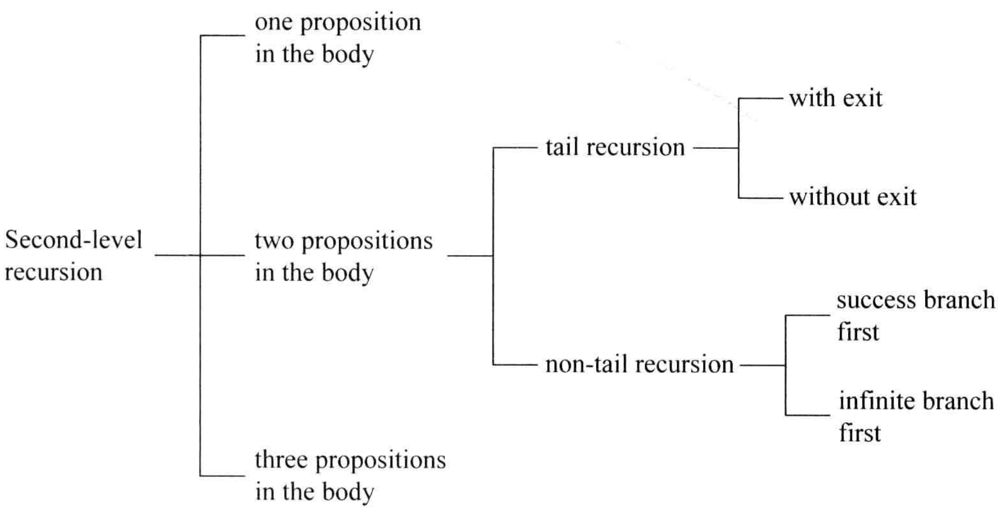


Fig. 14.8 Classification of second-level recursions

We can draw the following conclusions.

(1) In second-level tail recursion, infinite branch will not occur. Because the first empirical or mathematical connection operator in the body is not a recursive one, when it unifies unsuccessfully, control cannot go on.

(2) In second-level non-tail recursion, infinite branch is bound to occur. Because the first empirical or mathematical connection operator in the body is a recursive one, recursion can happen between the head and it repetitively.

(3) For second-level non-tail recursion, if for every empirical or mathematical connection operator, there is only one candidate in the conditional clauses, then the success branch is prior to the infinite branch. For example, in Example 14.7, ==>>only has (3) as its candidate. Its every unification is with the head of (3). If there is a solution, then control can get it. If for some empirical or mathematical connection operator, there is more than one candidates in the conditional clauses, then the infinite branch is prior to the success branch. Because second-level single quasi-Prolog adopts top-down searching strategy, every unification is with the top candidate, no chance with the other candidates, therefore, control runs into infinite branch at the beginning. For example, in Example 14.8, ==has two candidates: (3) and (4), unification is always with (3), no chance with (4), therefore, control runs into infinite branch at the beginning.

(4) The performance of the four second-level recursion with two propositions in the body is: second-level tail recursion with exit is the best, because it can be transformed into second-level iteration; second-level tail recursion without exit is the second, because it does not have infinite branch; success branch first non-tail recursion is the third, because it get the solution before it runs into infinite branch; infinite branch first non-tail recursion is the worst, but it is still practical, because control can detect the infinite branch and cut it.

Part 5
Mutually-inversistic granular computing

Mutually-inversistic granular computing includes mutually-inversistic fuzzy logic based granular computing and mutually-inversistic rough set based granular computing. Mutually-inversistic fuzzy logic based granular computing integrates mutually-inversistic logic with such branches of granular computing as fuzzy logic, rough set, interval, quotient space. In mutually-inversistic rough set, rough set is established on the basis of mutually-inversistic set theory.

Chapter 15 Mutually-inversistic fuzzy logic based granular computing

Mutually-inversistic fuzzy logic based granular computing includes mutually-inversistic fuzzy logic, mutually-inversistic rough fuzzy logic, mutually-inversistic fuzzy quotient space, mutually-inversistic fuzzy interval logic, mutually-inversistic rough fuzzy interval logic, mutually-inversistic fuzzy interval quotient space.

15.1 Mutually-inversistic fuzzy logic

15.1.1 Introduction to mutually-inversistic fuzzy logic

There are many ways of coping with uncertainty: Bayesian theory, Dempster-Shafer theory, fuzzy logic. Each method has its own shortcoming. Bayesian theory requires a priori probability which is hard to obtain. Dempster-Shafer theory has combinatorial explosion problem, and it is difficult to deal with infinitesimal belief. Fuzzy logic is insensitive to changes in parameters, and is subjective. Therefore, no method is generally recognized as the best method. Mutually-inversistic fuzzy logic provides people with a new choice of coping with uncertainty.

In mutually-inversistic fuzzy logic, Real numbers in the closed interval of [0, 1] are used to represent the degree of truth (membership grade) of propositions. 1 represents absolute truth, 0 absolute falsity, 0.5 "need not determine whether it is true or false". So, mutually-inversistic logic is a special case of mutually-inversistic fuzzy logic.

15.1.2 Adequate set of fuzzy logical operators

The adequate set of mutually-inversistic logic contains three logical operators: $\neg$, $\wedge$, and $\leqslant^{-1}$. The other logical operators can be expressed by the three logical operators.

$A \vee B$ can be defined as $\neg\{\neg A \wedge \neg B\}$,

$A \oplus B$ can be defined as $A \wedge \neg B \vee \neg A \wedge B$,

$A =^{-1} B$ can be defined as $\{A \leqslant^{-1} B\} \wedge \{B \leqslant^{-1} A\}$,

$A <^{-1} B$ can be defined as $\{A \leqslant^{-1} B\} \wedge \neg\{B \leqslant^{-1} A\}$,

$A / \wedge^{-1} B$ can be defined as $\neg\{A \leqslant^{-1} \neg B\}$,

$A \vee /^{-1} B$ can be defined as $\neg A \leqslant^{-1} B$,

$A \oplus^{-1} B$ can be defined as $\neg A =^{-1} B$,

$A \vee^{-1} B$ can be defined as $\neg A <^{-1} B$,

$A \times^{-1} B$ can be defined as $\{\neg A/\wedge^{-1}\neg B\} \wedge \{A/\wedge^{-1}\neg B\} \wedge \{\neg A/\wedge^{-1} B\} \wedge \{A/\wedge^{-1} B\}$.

Since mutually-inversistic fuzzy logic is a generalization of mutually-inversistic logic, the adequate set of mutually-inversistic fuzzy logic contains three fuzzy logical operators: $\neg_f$, $\wedge_f$, and $\leqslant_f^{-1}$.

15.1.3 Definitions of fuzzy logical operators

15.1.3.1 Definition of $\neg_f$

Suppose $a \in [0, 1]$, then

$$\neg_f a = 1 - a \tag{15.1}$$

The diagram of $\neg_f$ is shown in Fig. 15.1.

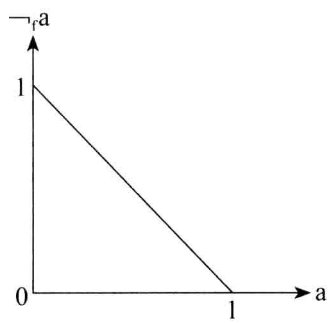


Fig. 15.1 The diagram for $\neg_f$

In mutually-inversistic logic, we have: $\neg F = T$, $\neg T = F$. In mutually-inversistic fuzzy logic, we have: $\neg_f 0 = 1$, $\neg_f 1 = 0$. Therefore, $\neg$ is a special case of $\neg_f$.

15.1.3.2 Definition of $\wedge_f$

Just as we can use two-dimensional contour map to represent three-dimensional topographic map, we can use two-dimensional equi-truth value line diagram to represent three-dimensional function $c = a \wedge_f b$. The equi-truth value line diagram is shown in Fig. 15.2.

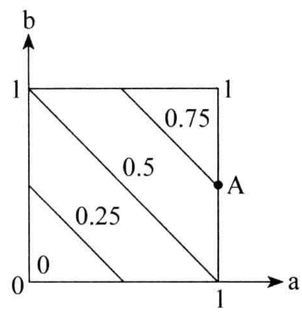


Fig. 15.2 Equi-truth value line diagram for $\wedge_f$

There are infinitely many equi-truth value lines in Fig. 15.2, we only draw a few of them. Each equi-truth value line in Fig. 15.2 can be written as

$$b=-a+intc, \tag{15.2}$$

where *intc* is the intercept of the equi-truth value line on the *b* axis.

Formula (15.2) can also be written as

$$a+b=intc. \tag{15.3}$$

According to Fig. 15.2, we have

$$a\wedge_f b=intc/2. \tag{15.4}$$

Suppose when $a=1$, $b=0.5$, we obtain the point A, the intercept on the *b* axis of the equi-truth value line of which is 1.5, according to (15.4), we obtain $a\wedge_f b=0.75$.

Substituting (15.3) into (15.4), we have

$$a\wedge_f b=(a+b)/2. \tag{15.5}$$

A and *b* can have weight:

$$a\wedge_f b=(w_a a+w_b b)/(w_a+w_b). \tag{15.6}$$

Conjuncts can be extended from two to n:

$$\wedge_f(a_1, a_2, \ldots, a_n)=(a_1+a_2+\ldots+a_n)/n. \tag{15.7}$$

In mutually-inversistic logic, we have: T$\wedge$T=T, F$\wedge$F=F. In mutually-inversistic fuzzy logic, we have: $1\wedge_f 1=(1+1)/2=1$, $0\wedge_f 0=(0+0)/2=0$. These two operations of $\wedge$ are the special cases of $\wedge_f$. In mutually-inversistic logic, we have: F$\wedge$T=F, T$\wedge$F=F. While in mutually-inversistic fuzzy logic, we have: $0\wedge_f 1=(0+1)/2=0.5$, $1\wedge_f 0=(1+0)/2=0.5$. These two operations of $\wedge$ are not the special cases of $\wedge_f$.

15.1.3.3 Definition of $\leqslant_f^{-1}$

"if A(x) is v, then C(x) is w" where v and w are fuzzy concepts is a fuzzy mutually inverse implication proposition or a fuzzy association rule, denoted as $\underset{\sim}{A}\leqslant_f^{-1}\underset{\sim}{C}$ where $\underset{\sim}{A}$ and $\underset{\sim}{C}$ are fuzzy sets. For example, "if x study English every day longer, then x's English exam score is better" where longer and better are fuzzy concepts is a fuzzy mutually inverse implication proposition or a fuzzy association rule, denoted as $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$ where $\underset{\sim}{T}$ and $\underset{\sim}{S}$ are fuzzy sets. X in "if A(x) is v, then C(x) is w" ranges over n training instances $\{e_1, \ldots, e_i, \ldots e_n\}$ and m test instances $\{e_{n+1}, \ldots, e_j, \ldots e_m\}$. The membership grade of e_i to $\underset{\sim}{A}$ is a_i, to $\underset{\sim}{C}$ is c_i, a_i, $c_i\in[0, 1]$ where 1 denotes absolute truth, 0 denotes absolute falsity. From a_i and c_i, we can determine the strength of support ss_i of e_i to $\underset{\sim}{A}\leqslant_f^{-1}\underset{\sim}{C}$, $ss_i\in[0, 1]$ where 1 denotes absolute support, 0 denotes absolute opposition, 0.5 denotes neutral or "need not determine whether it is true or false". For example, x in "if x study English every day longer, then x's English exam score is better" ranges over 4 training instances {Bob, Max, Pat, Sam} and 3 test instances {Ted, John, Alice}. Bob studies English 3 hours every day. 5 hours are regarded as long enough, 1 hour is regarded as short enough, so, the membership grade of Bob's 3 hours to $\underset{\sim}{T}$ is 0.5.

Bob scored 90 points in the English exam. In Chinese exam system, if one scored 60 points, then one passed the exam. And 100 points is the full mark. So, the membership grade of Bob's 90 points to $\underset{\sim}{S}$ is 0.75. From the membership grades of 0.5 and 0.75 we can determine Bob's strength of support ss_{Bob} to $\underset{\sim}{T} \leqslant_f^{-1} \underset{\sim}{S}$ is 0.75.

From a_i and c_i to determine ss_i is a binary function: $ss_i=f(a_i, c_i)$. Just as we can use contour lines to depict a three-dimensional topography in a two-dimensional plane, we can use equi_strength_of_support lines to depict the three-dimensional function $ss_i=f(a_i, c_i)$ in the two-dimensional plane. The diagram of equi_strength_of_support line is shown in Fig. 15.3.

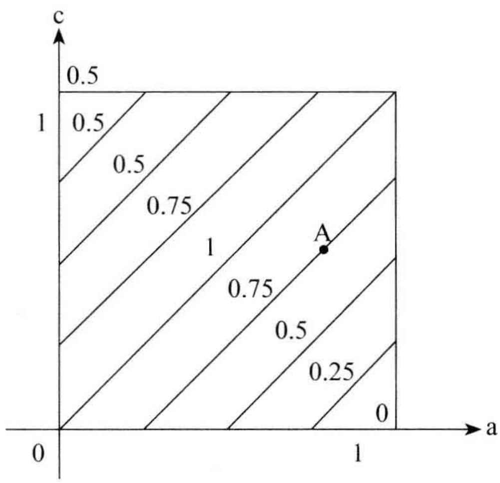


Fig.15.3 Diagram of equi_strength_of_support line

There are infinitely many equi_strength_of_support lines in Fig. 15.3, we only draw a few of them. The first, second, third, and fourth row of Table 1.2 correspond to the bottom-left, top-left, bottom-right, and top-right corner of Fig. 15.3, therefore, the inductive composition of $\leqslant^{-1}$ is a special case of the inductive composition of $\leqslant_f^{-1}$. According to Fig. 15.3, if we know a and c then we can determine ss. For example, suppose a=0.75, c=0.5, that is the point A in Fig. 15.3, which is on the equi_strength_of_support line 0.75, so, ss=0.75.

Every equi_strength_of_support line in Fig. 15.3 can be written as a slope-intercept equation:

$$c=a+intc. \tag{15.8}$$

From (15.8) we can obtain:

$$intc=c-a. \tag{15.9}$$

Fig. 15.3 depicts the relationship among a, c, and ss. But by (15.9), from a and c we can determine $intc$. So, according to Fig. 15.3, we can express the relationship between $intc$ and ss as a piecewise function:

$$ss=\begin{cases} 0.5, & 0.5\leqslant intc\leqslant 1 \quad \text{(neutral area)} \\ 1-intc, & 0\leqslant intc<0.5 \quad \text{(support area)} \\ 1+intc, & -0.5<intc<0 \quad \text{(support area)} \\ 1+intc, & intc=-0.5 \quad \text{(neutral area)} \\ 1+intc. & -1\leqslant intc<-0.5 \quad \text{(opposition area)} \end{cases} \tag{15.10}$$

Suppose a=0.75, c=0.5, then by (15.9), $intc$=0.75−0.5=0.25. *Intc* falls into the second row of (15.10), so, ss=1−0.25=0.75.

After ss_1, …, ss_i, …, ss_n are all computed, then we can mine the total strength of support ma of the fuzzy association rule $\underset{\sim}{A}\leqslant_f^{-1}\underset{\sim}{C}$:

$$ma=(ss_1+\ldots+ss_i+\ldots+ss_n)/n. \tag{15.11}$$

After ma is mined, it serves as the major premise of fuzzy hypothetical inference. The membership grade a_j of the test instance e_j to $\underset{\sim}{A}$ is the minor premise. According to the affirmative expression of fuzzy hypothetical inference:

$$co=ma*mi, \tag{15.12}$$

we can compute the conclusion co. When ma=1 and mi=1, according to (15.12), co=1∗1=1. That is, Table 1.3 is a special case of (15.12).

The membership grade c_j of the test instance e_j to $\underset{\sim}{C}$ is the conclusion. And the negative expression of fuzzy hypothetical inference is:

$$1-mi=ma*(1-co). \tag{15.13}$$

Both sides of (15.13) are subtracted by 1, then (15.14) is obtained:

$$mi=1-ma*(1-co). \tag{15.14}$$

When ma=1 and co=0, according to (15.14), mi=1−1∗(1−0)=0. That is, Table 1.4 is a special case of (15.14).

15.1.4 An example

Example 15.1: Suppose we have English study databases shown in Tables 15.1 and 15.2, and we want to do (1) fuzzy association rule mining; (2) machine inference.

Table 15.1 Training set of English study database

Universe U	English study time every day T (hours)	Attention A (%)	English exam score S (points)
Bob	3	100	90
Max	4	75	100
Pat	2	25	70
Sam	3	0	60

Table 15.2 Test set of English study database

Universe U	English study time every day T (hours)	Attention A (%)	English exam score S (points)
Ted	5	100	100
John	3	50	80
Alice	1	0	60

(1) Fuzzy association rule mining.

We want to mine the fuzzy association rule "if *x* study English every day longer, and *x*'s attention is focused, then *x*'s English exam score is better", denoted as $\underset{\sim}{T} \wedge_f \underset{\sim}{A} \leqslant_f^{-1} \underset{\sim}{S}$.

Longer, focused, and better are fuzzy concepts, their membership functions are shown in Figs. 15.4 to 15.6.

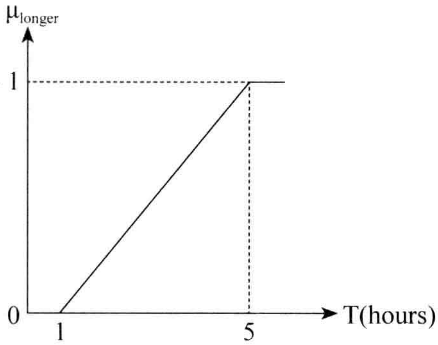


Fig. 15.4 Membership function for longer

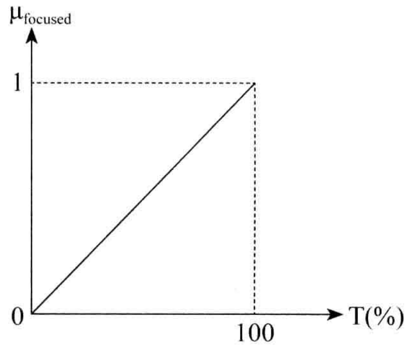


Fig. 15.5 Membership function for focused

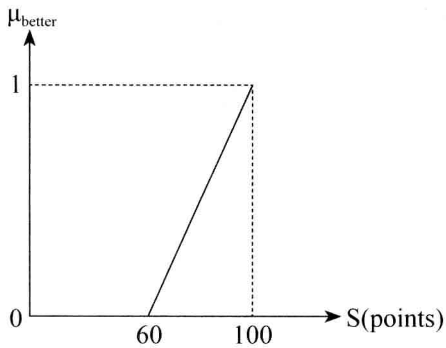


Fig. 15.6 Membership function for better

The fuzzy association rule mining algorithm is shown in Fig. 15.7.

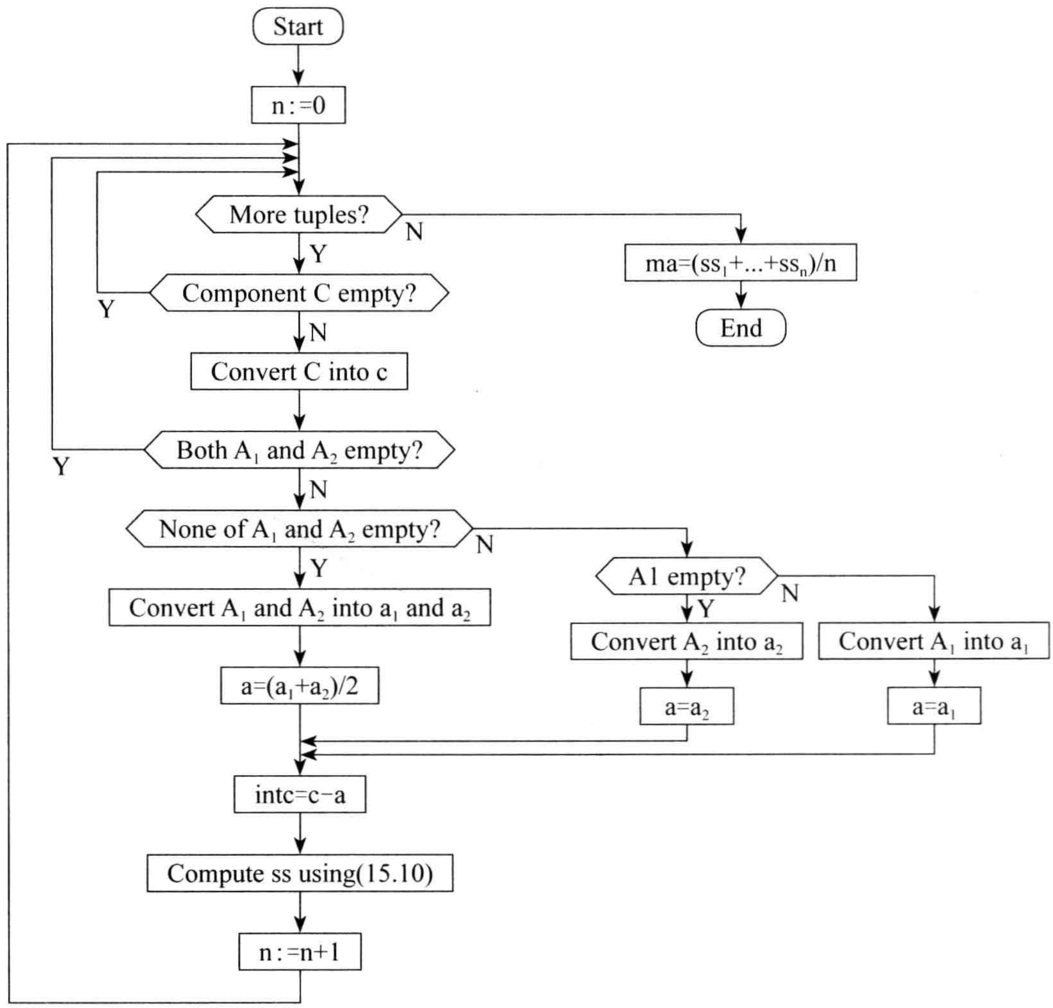


Fig. 15.7 Fuzzy association rule mining algorithm

According to Fig. 15.7, the algorithm uses the data in the training set. For Bob, S(Bob)=90 points, using Fig. 15.6, the algorithm converts 90 points into $c_1=0.75$; T(Bob)=3 hours, using Fig.15.4, the algorithm converts 3 hours into $a_{11}=0.5$; A(Bob)=100%, using Fig.15.5, the algorithm converts 100% into $a_{12}=1$. $A_1=(a_{11}+a_{12})/2=(0.5+1)/2=0.75$. $Intc_1=c_1-a_1=0.75-0.75=0$. $Intc_1$ falls into the second row of (15.10). $Ss_1=1-intc_1=1-0=1$. Likewise, for Max, $ss_2=0.75$; for Pat, $ss_3=1$; for Sam, $ss_4=0.75$. $Ma=(ss_1+ss_2+ss_3+ss_4)/4=(1+0.75+1+0.75)/4=0.875$. So, the total strength of support of $\underset{\sim}{T}\wedge_f\underset{\sim}{A}\leqslant_f^{-1}\underset{\sim}{S}$ is 0.875.

(2) Machine inference.

Now, we use the data in the test set. For Ted, T(Ted)=5 hours, using Fig.15.4, we convert 5 hours into $a_{51}=1$; A(Ted)=100%, using Fig. 15.5, we convert 100% into $a_{52}=1$; $Mi_5=(a_{51}+a_{52})/2=(1+1)/2=1$. $Co_5=ma*mi_5=0.875*1=0.875$. Using Fig. 15.6, we convert $co_5=0.875$ back into points, and we estimate Ted's score is $60+40*0.875=95$

points, which is near to his real score 100 points. Likewise, for John, $mi_6=0.5$, $co_6=ma*mi_6=0.875*0.5=0.4375$, John's estimated score is 77.5, which is near to his real score 80; for Alice, $mi_7=0$, $co_7=ma*mi_7=0.875*0=0$, Alice's estimated score is 60, which equals to her real score 60.

15.1.5 Conclusions on mutually-inversistic fuzzy logic

We can draw the following conclusions:

(1) In fuzzy logic, a membership function is subjective, while in mutually-inversistic fuzzy logic, a membership function is objective. Take Fig. 15.6 as an example. In Chinese exam system, if one does not get 60 points, then one fails the exam; if one gets 100 points, then one gets full mark. So, 60 points is the worst score (membership grade 0), 100 points is the best score (membership grade 1). Draw a straight line between (60, 0) and (100, 1), we obtain the membership function of better.

(2) In mutually-inversistic fuzzy logic, in Fig. 15.3, the equi_strength_of_support line of strength of support 1 is on the main diagonal, and the membership functions such as Figs. 15.4 to 15.6 are proportional lines. This means for relatively true $mi_5=1$, relatively neutral $mi_6=0.5$, relatively false $mi_7=0$, one fuzzy association rule $\underset{\sim}{I} \wedge_f \underset{\sim}{A} \leqslant_f^{-1} \underset{\sim}{S}$ and its membership grade ma=0.875 is enough. While in fuzzy logic, corresponding to Fig. 15.6, we have three membership functions. For better score, we have Fig. 15.8, for plain score, we have Fig. 15.9, for worse score, we have Fig. 15.10. The same is true for Figs. 15.4 and 15.5. Therefore, in fuzzy logic, we need three fuzzy association rules. For relatively true mi, we need "if x study English every day longer, and x's attention is more focused, then x's English exam score is better". For relatively neutral mi, we need "if x study English every day not long and not short, and x's attention is medium, then x's English exam score is plain". For relatively false mi, we need "if x study English every day shorter, and x's attention is less focused, then x's English exam score is worse".

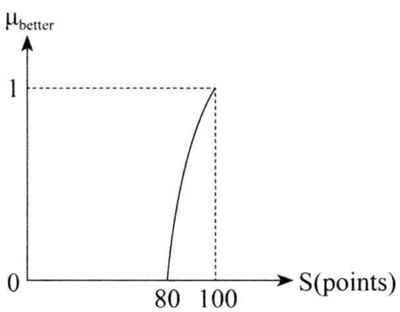


Fig. 15.8 Membership function for better

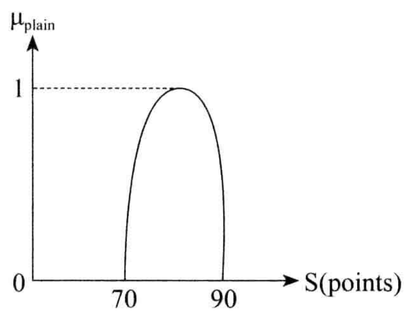


Fig. 15.9 Membership function for plain

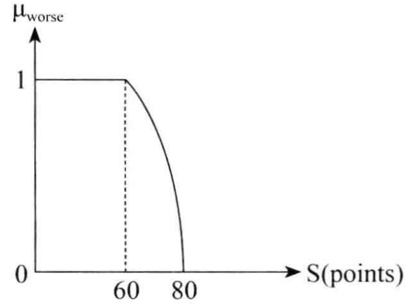


Fig. 15.10 Membership function for worse

(3) Fuzzy logic is insensitive to the changes in parameter. While mutually-inversistic fuzzy logic is sensitive to the changes in parameter. From Fig. 15.3 we see that *ss* changes if *a* or *c* changes. From (15.5) we see that $a\wedge_f b$ changes if *a* or *b* changes. From (15.12) we see that co changes if ma or mi changes.

15.2 Mutually-inversistic rough fuzzy logic

Mutually-inversistic rough fuzzy logic is the integration of mutually-inversistic fuzzy logic and rough fuzzy sets.

15.2.1 Rough fuzzy sets

Rough fuzzy set is the integration of rough set and fuzzy set. It introduces equivalence relation into the definition of fuzzy set, defines membership grade on equivalence classes, and objects in the same equivalence class have the same membership grade. Its definition is as follows.

Suppose U is a given universe, $\underset{\sim}{A}$ is a fuzzy set, R is an equivalence relation on U, the partition R corresponds is $U/R=\{U_1, U_2, \ldots, U_n\}$, then the lower and upper approximations of fuzzy set $\underset{\sim}{A}$ under the equivalence relation R are:

$\mu\underline{R}\underline{\underset{\sim}{A}}(U_i)=\inf\{\mu\underset{\sim}{A}(u):u\in U_i\}$,

$\mu R\underset{\sim}{A}(U_i)=\sup\{\mu\underset{\sim}{A}(u):u\in U_i\}$,

respectively.

15.2.2 Mutually-inversistic rough fuzzy logic

Example 15.2: Suppose we have English study databases shown in Tables 15.1 and 15.2. We want to: (1) use the attribute A as the equivalence relation, and mine the lower and upper approximations of the fuzzy association rule on equivalence classes "if *x* study English every day longer, then *x*'s English exam score is better" denoted as $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$. (2) make machine inference.

(1) Fuzzy association rule mining.

Now, we use the data in the training set. The attribute A is used as the equivalence relation. A$\geqslant$50% is the equivalence class Attention={Bob, Max}, A$<$50% is the equivalence class Inattention={Pat, Sam}. Longer and better are fuzzy concepts, their membership functions are shown in Figs. 15.4 and 15.6. Converting the attribute T in Table 15.1 by Fig. 15.4, we obtain the fuzzy set $\underset{\sim}{T}$:

$\underset{\sim}{T}$=0.5/Bob+0.75/Max+0.25/Pat+0.5/Sam.

The lower and upper approximations of $\underset{\sim}{T}$ are $\underline{A}\underset{\sim}{T}$ and $\overline{A}\underset{\sim}{T}$ respectively:

$\underline{A}\underset{\sim}{T}$=0.5/Bob+0.5/Max+0.25/Pat+0.25/Sam,

$\overline{A}\underset{\sim}{T}$=0.75/Bob+0.75/Max+0.5/Pat+0.5/Sam,

$\mu\underline{A}\underset{\sim}{T}$(Attention)=0.5,

$\mu\underline{A}\underset{\sim}{T}$(Inattention)=0.25,

$\mu\overline{A}\underset{\sim}{T}$(Attention)=0.75,

$\mu\overline{A}\underset{\sim}{T}$(Inattention)=0.5.

Likewise,

$\underset{\sim}{S}$=0.75/Bob+1/Max+0.25/Pat+0/Sam,

$\underline{A}\underset{\sim}{S}$=0.75Bob+0.75/Max+0/Pat+0/Sam,

$\overline{A}\underset{\sim}{S}$=1/Bob+1/Max+0.25/Pat+0.25/Sam,

$\mu\underline{A}\underset{\sim}{S}$(Attention)=0.75,

$\mu\underline{A}\underset{\sim}{S}$(Inattention)=0,

$\mu\overline{A}\underset{\sim}{S}$(Attention)=1,

$\mu\overline{A}\underset{\sim}{S}$(Inattention)=0.25.

Now, let us mine the lower approximation of the fuzzy association rule $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$ on equivalence class Attention.

Intc=$\mu\underline{A}\underset{\sim}{S}$(Attention)−$\mu\underline{A}\underset{\sim}{T}$(Attention)=0.75−0.5=0.25, intc falls into the second row of (15.10), and ss=1−intc=1−0.25=0.75. Likewise, the ss's of the upper approximation of $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$ on Attention, the lower approximation of $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$ on Inattention, the upper approximation of $\underset{\sim}{T}\leqslant_f^{-1}\underset{\sim}{S}$ on Inattention are all 0.75.

(2) Machine inference.

Now, we use the data in the test set. T(Ted)=5 hours. Using Fig. 15.4, we convert 5 hours into mi_5=1. Ted falls into the equivalence class Attention. Co_5=ma∗mi_5=0.75∗1=0.75. Using Fig. 15.6, we convert co_5=0.75 back to points, and we estimate that Ted's score is 60+40∗0.75=90, which is not near to his real score 100.

15.2.3 Conclusions on mutually-inversistic rough fuzzy logic

We can draw the following conclusions.

The advantage of mutually-inversistic rough fuzzy logic over mutually-inversistic

fuzzy logic is that in the latter, we have to mine for every object, while in the former, we need only to mine once for the entire equivalence class. The disadvantage of mutually-inversistic rough fuzzy logic over mutually-inversistic fuzzy logic is that in the former, the attribute of the equivalence relation does not included in the fuzzy association rule, what we mine is "if x study English every day longer, then x's English exam score is better", and the ss is low, only 0.75. While in the latter, what we mine is "if x study English every day longer, and x's attention is focused, then x's English exam score is better", and the ss is high; i.e., 0.875.

15.3 Mutually-inversistic fuzzy quotient space

Mutually-inversistic fuzzy quotient space is the integration of mutually-inversistic fuzzy logic and fuzzy quotient spaces.

15.3.1 Fuzzy quotient space

Suppose U is the universe, $\underset{\sim}{A}$ is a fuzzy set on U, [U] is the quotient space of U, $[\underset{\sim}{A}]$ is the quotient fuzzy set induced on [U], then

$\mu_{[\underset{\sim}{A}]}([u])=\max\{\mu_{\underset{\sim}{A}}(u)|u\in[u]\}$, where $[u]\in[U]$

is the quotient membership grade of the quotient fuzzy set.

15.3.2 Mutually-inversistic fuzzy quotient space

Example 15.3: Suppose we have English study databases shown in Tables 15.1 and 15.2, and we want to (1) mine the fuzzy association rule "if x study English every day longer, and x's attention is focused, then x's English exam score is better", denoted as $\underset{\sim}{T}\wedge_f\underset{\sim}{A}\leqslant_f^{-1}\underset{\sim}{S}$ on the equivalence classes. (2) make machine inference.

(1) Fuzzy association rule mining.

We use the data in the training set.

We use attribute A to classify U. A$\geqslant$50% is one equivalence class, A$<$50% is the other one, obtaining the quotient space of U: {(Bob, Max), (Pat, Sam)} or $\{[\text{Bob}]_A, [\text{Pat}]_A\}$.

Longer, focused, and better are fuzzy concepts, their membership functions are shown in Figs. 15.4 to 15.6. Using Fig. 15.4 to convert the attribute T in Table 15.1 into membership grades, we obtain the fuzzy set $\underset{\sim}{T}$:

$\underset{\sim}{T}$=0.5/Bob+0.75/Max+0.25/Pat+0.5/Sam,

and

$\mu_{[\underset{\sim}{T}]}([\text{Bob}]_A)=0.75$,

$\mu_{[\underset{\sim}{T}]}([\text{Pat}]_A)=0.5$.

Likewise,

$\underset{\sim}{A}$=1/Bob+0.75/Max+0.25/Pat+0/Sam,

$\mu_{[\underset{\sim}{A}]}([\text{Bob}]_A)=1$,

$\mu_{[\underset{\sim}{A}]}([\text{Pat}]_A)=0.25$.

$\underset{\sim}{S}=0.75/\text{Bob}+1/\text{Max}+0.25/\text{Pat}+0/\text{Sam}$,

$\mu_{[\underset{\sim}{S}]}([\text{Bob}]_A)=1$,

$\mu_{[\underset{\sim}{S}]}([\text{Pat}]_A)=0.25$.

For equivalence class $[\text{Bob}]_A$, we have $a=(\mu_{[\underset{\sim}{T}]}([\text{Bob}]_A)+\mu_{[\underset{\sim}{A}]}([\text{Bob}]_A))/2=(0.75+1)/2=0.875$, $c=\mu_{[\underset{\sim}{S}]}([\text{Bob}]_A)=1$, intc=c−a=1−0.875=0.125. Intc falls into the second row of (15.10), and ss=1−intc=1−0.125=0.875.

Likewise, for equivalence class $[\text{Pat}]_A$, ss=0.875.

(2) Machine inference.

Now, let us use the data in the test set. For Ted, $a_{51}=1$, $a_{52}=1$, $mi_5=(a_{51}+a_{52})/2=(1+1)/2=1$. Ted belongs to the equivalence class $[\text{Bob}]_A$, and $co_5=ma*mi_5=0.875*1=0.875$. Using Fig. 15.6 to convert 0.875 back to points, we estimate that Ted's score is 60+40*0.875=95, which is near his real score 100.

15.3.3 Conclusions on mutually-inversistic fuzzy quotient space

In mutually-inversistic fuzzy logic, we have to mine for every object. While in mutually-inversistic fuzzy quotient space, we need only to mine once for the entire equivalence class. In mutually-inversistic rough fuzzy logic, what we mine is "if *x* study English every day longer, then *x*'s English exam score is better", and ss's are low, only 0.75. While in mutually-inversistic fuzzy quotient space, what we mine is "if *x* study English every day longer, and *x*'s attention is focused, then *x*'s English exam score is better", and ss's are high; i.e., 0.875. Therefore, mutually-inversistic fuzzy quotient space is the best of the three logics.

15.4 Mutually-inversistic fuzzy interval logic

Mutually-inversistic fuzzy interval logic is the integration of mutually-inversistic fuzzy logic and interval arithmetic.

15.4.1 Interval arithmetic

Suppose $[x^-, x^+]$ denotes an interval, where x^- is the lower limit, x^+ is the upper limit of the interval. The arithmetic operations of intervals are as follows.

$[x_1^-, x_1^+]+[x_2^-, x_2^+]=[x_1^-+x_2^-, x_1^++x_2^+]$,

$[x_1^-, x_1^+]-[x_2^-, x_2^+]=[x_1^--x_2^+, x_1^+-x_2^-]$,

$[x_1^-, x_1^+]*[x_2^-, x_2^+]=[\min(x_1^-*x_2^-, x_1^-*x_2^+, x_1^+*x_2^-, x_1^+*x_2^+), \max(x_1^-*x_2^-, x_1^-*x_2^+, x_1^+*x_2^-, x_1^+*x_2^+)]$,

$1/[x_1^-, x_1^+]=[1/x_1^+, 1/x_1^-]$.

15.4.2 Mutually-inversistic fuzzy interval logic

The membership grade of an interval cannot be described by a single value, but by a membership grade interval. Suppose the membership grade interval of the interval $[x_i^-, x_i^+]$ is $[\mu x_i^-, \mu x_i^+]$, then we use

$${}^{-}\underset{\sim}{A}^{+}=[\mu x_1^-, \mu x_1^+]/[x_1^-, x_1^+]+\ldots+[\mu x_i^-, \mu x_i^+]/[x_i^-, x_i^+]+\ldots+[\mu x_n^-, \mu x_n^+]/[x_n^-, x_n^+]$$

to denote the fuzzy interval ${}^{-}\underset{\sim}{A}^{+}$.

$\neg_f[\mu x^-, \mu x^+]$ is defined as $[1-\mu x^+, 1-\mu x^-]$. $[\mu x_1^-, \mu x_1^+]\wedge_f[\mu x_2^-, \mu x_2^+]$ is defined as $[(\mu x_1^- +\mu x_2^-)/2, (\mu x_1^+ +\mu x_2^+)/2]$.

As to the fuzzy interval mutually inverse implication proposition (fuzzy interval association rule) ${}^{-}\underset{\sim}{A}^{+}\leqslant_f^{-1}{}^{-}\underset{\sim}{C}^{+}$, there is a process of mining it, and after it is mined, there is a process of using it as the major premise to make fuzzy hypothetical inference.

Fuzzy interval association rule mining is as follows: Suppose the universe is $\{[e_1^-, e_1^+], \ldots, [e_i^-, e_i^+], \ldots, [e_n^-, e_n^+]\}$. Suppose the membership interval of $[e_i^-, e_i^+]$ to ${}^{-}\underset{\sim}{A}^{+}$ is $[a_i^-, a_i^+]$, to ${}^{-}\underset{\sim}{C}^{+}$ is $[c_i^-, c_i^+]$. We use Algorithm 15.1 to compute the strength of support interval $[ss_i^-, ss_i^+]$. After $[ss_1^-, ss_1^+],\ldots,[ss_n^-, ss_n^+]$ are all computed, we use $[ss_t^-, ss_t^+]=([ss_1^-, ss_1^+]+\ldots+[ss_n^-, ss_n^+])/n$ to compute the total strength of support $[ss_t^-, ss_t^+]$.

$[ss_i^-, ss_i^+]$ is computed as follows: Let

$[intc_i^-, intc_i^+]=[c_i^-, c_i^+]-[a_i^-, a_i^+]$,

that is,

$intc_i^-=c_i^- -a_i^+$,

$intc_i^+=c_i^+ -a_i^-$.

After $[intc_i^-, intc_i^+]$ is computed, we can obtain $[ss_i^-, ss_i^+]$ according to Fig. 15.3 and (15.10). The algorithm from $[a_i^-, a_i^+]$ and $[c_i^-, c_i^+]$ to $[ss_i^-, ss_i^+]$ is shown in Algorithm 15.1.

Algorithm 15.1

(1) $intc^-=c^- -a^+$;

(2) $intc^+=c^+ -a^-$;

(3) if ($intc^-\leqslant 0\leqslant intc^+$)/*[$intc^-$, $intc^+$] span both sides of the main diagonal of Fig. 15.3*/

(4) then {$ss^+=1$;

(5) if($intc^+\geqslant 0.5$) then $temp_1=0.5$;

(6) else $temp_1=1-intc^+$;

(7) $temp_2=1+intc^-$;

(8) If($temp_2\leqslant temp_1$) then $ss^-=temp_2$;

(9) else $ss^-=temp_1$;

(10) }

(11) else if($intc^->0$)/*[$intc^-$, $intc^+$] is on the top-left corner of Fig. 15.3*/

```
      then {if(intc^- ≥0.5)  then ss^-=ss^+=0.5;
                             else if(intc^+ ≥0.5)
                                  then {ss^-=0.5;
                                        ss^+=1-intc^-;
                                       }
                                  else {ss^+=1-intc^-;
                                        ss^-=1-intc^+;
                                       }
           }
else if(intc^+ <0)/*[intc^-, intc^+] is on the bottom-right corner of Fig. 15.3*/
      then {ss^+=1+intc^+;
            ss^-=1+intc^-;
           }
```

15.4.3 An example

Suppose we have an air quality interval-valued database shown in Table 15.3.

Table 15.3 Air quality interval-valued database

Universe X	Coal consumption CC	Traffic density TD	SO_2 density SO_2
$[x_1^-, x_1^+]$	[0.58, 0.62]	[2.5, 3.5]	[0.015, 0.025]
$[x_2^-, x_2^+]$	[2, 2.5]	[8.5, 9.5]	[0.085, 0.095]
$[x_3^-, x_3^+]$	[0.2, 0.3]	[18, 20]	[0.011, 0.015]
$[x_4^-, x_4^+]$	[1.35, 1.45]	[47.5, 48]	[0.08, 0.095]
$[x_5^-, x_5^+]$	[0.135, 0.142]	[10, 11.5]	[0.02, 0.03]

We use the first four tuples as the training set, the fifth tuple as the test set. We want to do (1) fuzzy interval association rule mining; (2) machine inference.

(1) Fuzzy interval association rule mining.

We want to mine "if coal consumption (CC) is high, and traffic density (TD) is heavy, then SO_2 density (SO_2) is dense", denoted as ${}^-\underset{\sim}{CC}^+ \wedge_f {}^-\underset{\sim}{TD}^+ \le_f^{-1} {}^-\underset{\sim}{SO_2}^+$. High, heavy, and dense are fuzzy concepts, their membership functions are shown in Figs. 15.11 to 15.13.

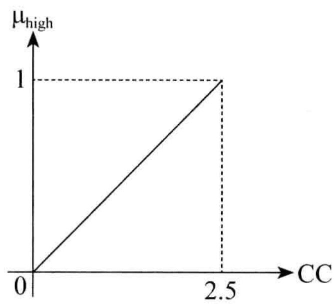


Fig. 15.11 Membership function for high

Using Fig. 15.11 to convert every interval in the attribute CC into membership grade interval, we obtain the fuzzy interval ${}^-\underset{\sim}{CC}^+$:

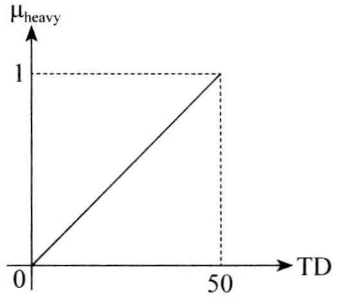


Fig. 15.12 **Membership function for heavy**

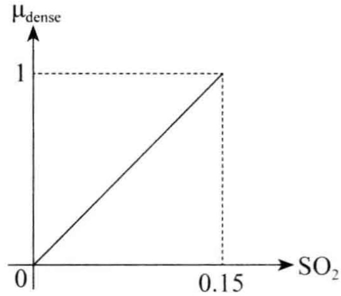


Fig. 15.13 **Membership function for dense**

$^{-}\underline{CC}^{+}$=[0.232, 0.248]/[x_1^-, x_1^+]+[0.8, 1]/[x_2^-, x_2^+]+[0.08, 0.12]/[x_3^-, x_3^+]+[0.54, 0.58]/[x_4^-, x_4^+]. (15.15)

Likewise,

$^{-}\underline{TD}^{+}$=[0.05, 0.07]/[x_1^-, x_1^+]+[0.17, 0.19]/[x_2^-, x_2^+]+[0.36, 0.4]/[x_3^-, x_3^+]+[0.95, 0.96]/[x_4^-, x_4^+], (15.16)

$^{-}\underline{SO_2}^{+}$=[0.1, 0.167]/[x_1^-, x_1^+]+[0.566, 0.633]/[x_2^-, x_2^+]+[0.073, 0.1]/[x_3^-, x_3^+]+[0.533, 0.633]/[x_4^-, x_4^+]. (15.17)

Now, let us compute the strength of support interval of [x_1^-, x_1^+] to $^{-}\underline{CC}^{+}\wedge_f {}^{-}\underline{TD}^{+}\leqslant_f^{-1}{}^{-}\underline{SO_2}^{+}$:

[a_1^-, a_1^+]=([0.232, 0.248]+[0.05, 0.07])/2=[(0.232+0.05)/2, (0.248+0.07)/2]=[0.141, 0.159].

[$intc_1^-$, $intc_1^+$]=[c_1^-, c_1^+]−[a_1^-, a_1^+]=[0.1, 0.167]−[0.141, 0.159]=[(0.1−0.159), (0.167−0.141)]=[−0.059, 0.026].

Ss_1^+=1. $Temp_{11}$=1−$intc_1^+$=1−0.026=0.974. $Temp_{21}$=1+$intc_1^-$=1+(−0.059)=0.941. Ss_1^-=$temp_{21}$ =0.941.

So, the strength of support interval of [x_1^-, x_1^+] to $^{-}\underline{CC}^{+}\wedge_f {}^{-}\underline{TD}^{+}\leqslant_f^{-1}{}^{-}\underline{SO_2}^{+}$ is [ss_1^-, ss_1^+] =[0.941, 1].

Likewise, the strength of support intervals of [x_2^-, x_2^+], [x_3^-, x_3^+], and [x_4^-, x_4^+] to $^{-}\underline{CC}^{+}\wedge_f {}^{-}\underline{TD}^{+}\leqslant_f^{-1}{}^{-}\underline{SO_2}^{+}$ are [0.952, 1], [0.823, 0.88], and [0.763, 0.888]. The total strength of support interval [ss_t^-, ss_t^+] of [x_1^-, x_1^+], [x_2^-, x_2^+], [x_3^-, x_3^+], and [x_4^-, x_4^+] to $^{-}\underline{CC}^{+}\wedge_f {}^{-}\underline{TD}^{+}\leqslant_f^{-1}{}^{-}\underline{SO_2}^{+}$ is:

[ss_t^-, ss_t^+]=([ss_1^-, ss_1^+]+[ss_2^-, ss_2^+]+[ss_3^-, ss_3^+]+[ss_4^-, ss_4^+])/4=([0.941, 1]+[0.952, 1]+[0.823, 0.88]+[0.763, 0.888])/4=[(0.941+0.952+0.823+0.763)/4, (1+1+0.88+0.888)/4]=[0.871, 0.942].

(2) Machine inference.

[CC_5^-, CC_5^+]=[0.135, 0.142], [TD_5^-, TD_5^+]=[10, 11.5]. Using Fig. 15.11 into fuzzy interval, we obtain:

$^{-}\underline{CC_5}^{+}$=[0.054, 0.057]/[x_5^-, x_5^+]. (15.18)

Likewise,

$\underset{\sim}{TD}_5^+=[0.2, 0.23]/[x_5^-, x_5^+]$. (15.19)

$[mi_5^-, mi_5^+]=([0.054, 0.057]+[0.2, 0.23])/2=[(0.054+0.2)/2, (0.057+0.23)/2]=[0.127, 0.144]$. (15.20)

$[co_5^-, co_5^+]=[ma^-, ma^+]*[mi_5^-, mi_5^+]=[ss_t^-, ss_t^+]*[mi_5^-, mi_5^+]=[0.871, 0.942]*[0.127, 0.144]=[0.871*0.127, 0.942*0.144]=[0.111, 0.135]$.

Using Fig. 15.13 to convert [0.111, 0.135] back to SO_2, we have $[SO_{25}^-, SO_{25}^+]=[0.111*0.15, 0.135*0.15]=[0.017, 0.02]$, which is not far from its actual interval [0.02, 0.03].

15.4.4 Conclusion on mutually-inversistic fuzzy interval logic

The conventional fuzzy interval association rule mining algorithm mines the interval-value as a single value, while mutually-inversistic fuzzy interval logic mines the strength of support intervals, more accurate than the former.

15.5 Mutually-inversistic rough fuzzy interval logic and mutually-inversistic fuzzy interval quotient space

Rough fuzzy set can be generalized to rough fuzzy interval. Mutually-inversistic rough fuzzy interval logic is the integration of mutually-inversistic fuzzy interval logic and rough fuzzy interval, just as mutually-inversistic rough fuzzy logic is the integration of mutually-inversistic fuzzy logic and rough fuzzy set. The construction of mutually-inversistic rough fuzzy interval logic is left to the reader.

Fuzzy quotient space can be generalized to fuzzy interval quotient space. Mutually-inversistic fuzzy interval quotient space is the integration of mutually-inversistic fuzzy interval logic and fuzzy interval quotient space, just as mutually-inversistic fuzzy quotient space is the integration of mutually-inversistic fuzzy logic and fuzzy quotient space. The construction of mutually-inversistic fuzzy interval quotient space is left to the reader.

15.6 Conclusions of mutually-inversistic fuzzy logic based granular computing

Mutually-inversistic fuzzy logic mines the fuzzy association rule on the finest granule object. Mutually-inversistic rough fuzzy logic and mutually-inversistic fuzzy quotient space mine it on the coarser granule equivalence class. Mutually-inversistic fuzzy interval logic mines it on the coarser granule interval. Mutually-inversistic rough fuzzy interval logic and mutually-inversistic fuzzy interval quotient space mine it on the coarsest granule equivalence class of interval.

Chapter 16 Mutually-inversistic rough set based granular computing

Mutually-inversistic rough set is formed by building rough set on the basis of mutually-inversistic set theory. Mutually-inversistic rough set based granular computing includes mutually-inversistic rough set, mutually-inversistic fuzzy rough set, mutually-inversistic mathematical morphology, mutually-inversistic evidence theory, mutually-inversistic point set topology, mutually-inversistic concept lattice, mutually-inversistic second-type covering rough set.

16.1 Mutually-inversistic rough set

16.1.1 Conventional rough set

Given a knowledgebase K=(U, R), where U is the universe, R is an equivalence relation on U, then for every $X \subseteq U$, the lower approximation and upper approximation of X with respect to R are:

$$\begin{aligned}\underline{R}(X) &= \{x \mid (\forall x \in U) \wedge ([x]_R \subseteq X)\}\\ &= \cup \{Y \mid (Y \in U/R) \wedge (Y \subseteq X)\},\\ \overline{R}(X) &= \{x \mid (\forall x \in U) \wedge ([x]_R \cap X \neq \varnothing)\}\\ &= \cup \{Y \mid (Y \in U/R) \wedge (Y \cap X \neq \varnothing)\}.\end{aligned}$$

Set $bn_R(X)=\overline{R}(X)-\underline{R}(X)$ is called the R boundary of X. If $bn_R(X)$ is not empty, then X is called a rough set. The lower and upper approximations and the boundary can be denoted by Fig. 16.1.

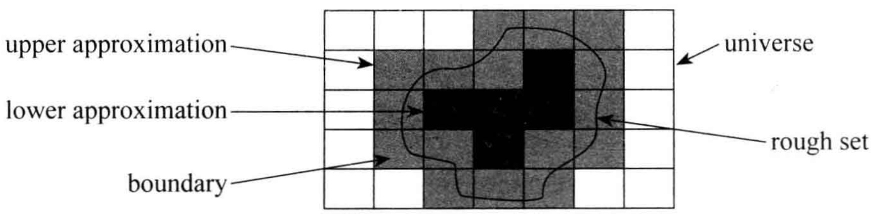


Fig. 16.1 Rough set

16.1.2 Mutually-inversistic rough set

Rough set can be constructed on mutually-inversistic set theory to form mutually-

inversistic rough set. The lower and upper approximations can be defined as:

$$\underline{R}(X)=\{x|(x\in U)\wedge([x]_R\subseteq^{-1}X)\}$$
$$=\cup\{Y|(Y\in U/R)\wedge(Y\subseteq^{-1}X)\},$$
$$\overline{R}(X)=\{x|(x\in U)\wedge([x]_R|\cap^{-1}X)\}$$
$$=\cup\{Y|(Y\in U/R)\wedge(Y|\cap^{-1}X)\}.$$

$P\subseteq^{-1}Q$ is a mutually inverse being contained connection, it is a decision rule in conventional rough set. $P|\cap^{-1}Q$ is a mutually inverse intersection connection. $\{P|\cap^{-1}Q\}\cap\sim\{P|\cap^{-1}\sim Q\}$ is a strongly mutually inverse intersection connection. $\{P|\cap^{-1}Q\}\cap\{P|\cap^{-1}\sim Q\}$ is a weakly mutually inverse intersection connection, it is a boundary in conventional rough set.

16.1.3 An example

Example 16.1: Suppose we have a college entrance exam knowledgebase shown in Table 16.1.

Table 16.1 College entrance exam knowledgebase

U	E	H	S
1	A	P	+
2	A	F	+
3	B	P	+
4	B	F	+
5	B	F	−
6	C	P	+
7	C	P	−
8	C	F	−
9	D	P	−
10	D	F	−

In Table 16.1, U denotes the universe, in which there are 10 examinees. E denotes the exam score, there are four kinds of scores: A, B, C, and D. H denotes health checkup, in which P denotes pass, F denotes fail. S denotes admission to college, in which+denotes admitted, - denotes not admitted.

According to Table 16.1, we can make indiscernible relation as follows:

$U/E=\{\{1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}, \{9, 10\}\}=\{E_A, E_B, E_C, E_D\}$,

$U/H=\{\{1, 3, 6, 7, 9\}, \{2, 4, 5, 8, 10\}\}=\{H_P, H_F\}$,

$U/S=\{\{1, 2, 3, 4, 6\}, \{5, 7, 8, 9, 10\}\}=\{S_+, S_-\}$.

16.1.3.1 Mutually inverse being contained connection and strongly mutually inverse intersection connection mining

We start from the lower approximations, take E and H as the conditional attributes, take S as the decision attribute, make knowledge reduction on Table 16.1, mine the following mutually inverse being contained connections:

$$E_A \subseteq^{-1} S_{+}, \tag{16.1}$$

$$E_B \cap H_P \subseteq^{-1} S_{+}, \tag{16.2}$$

$$E_C \cap H_F \subseteq^{-1} S_{-}, \tag{16.3}$$

$$E_D \subseteq^{-1} S_{-}. \tag{16.4}$$

From the mutually inverse being contained connection $P \subseteq^{-1} Q$ and (16.5), (16.6), using second-level hypothetical inference, we can infer the strongly mutually inverse intersection connection (16.7).

$$\{P \subseteq^{-1} Q\} \subseteq^{-1} \{P | \cap^{-1} Q\}, \tag{16.5}$$

$$\{P \subseteq^{-1} Q\} =^{-1} \sim \{ P | \cap^{-1} \sim Q\}, \tag{16.6}$$

$$\{P | \cap^{-1} Q\} \cap \sim \{ P | \cap^{-1} \sim Q\}. \tag{16.7}$$

For example, starting from (16.1), using (16.5) and (16.6), we can infer:

$$\{E_A | \cap^{-1} S_{+}\} \cap \sim \{ E_A | \cap^{-1} \sim S_{+}\}.$$

16.1.3.2 Single antecedent weakly mutually inverse intersection connection mining

$$\{P | \cap^{-1} Q\} \cap \{ P | \cap^{-1} \sim Q\}. \tag{16.8}$$

Formula (16.8) is a single antecedent weakly mutually inverse intersection connection, because there is only one antecedent P.

We can use upper approximations to mine single antecedent weakly mutually inverse intersection connections.

The S_+ upper approximation of E is

$$\overline{E}(S_+) = \cup \{E_i \in U/E : E_i | \cap^{-1} S_+\},$$

that is, the union of those E_i in E that intersect with S_+. Because $E_A | \cap^{-1} S_+$, $E_B | \cap^{-1} S_+$, $E_C | \cap^{-1} S_+$, $\sim \{E_D | \cap^{-1} S_+\}$,

$$\overline{E}(S_+) = E_A \cup E_B \cup E_C.$$

Likewise,

$$\overline{E}(S_-) = \overline{E}(\sim S_+) = E_B \cup E_C \cup E_D.$$

For E_i in $\overline{E}(S_+)$, we have $E_i | \cap^{-1} S_+$. For E_i in $\overline{E}(\sim S_+)$, we have $E_i | \cap^{-1} \sim S_+$. E_B and E_C belong to both $\overline{E}(S_+)$ and $\overline{E}(\sim S_+)$, therefore, we have

$$\{E_B | \cap^{-1} S_+\} \cap \{E_B | \cap^{-1} \sim S_+\}, \tag{16.9}$$

$$\{E_C | \cap^{-1} S_+\} \cap \{E_C | \cap^{-1} \sim S_+\}. \tag{16.10}$$

Likewise, we have

$$\{H_P|\cap^{-1}S_+\}\cap\{H_P|\cap^{-1}{\sim}S_+\}, \tag{16.11}$$

$$\{H_F|\cap^{-1}S_+\}\cap\{H_F|\cap^{-1}{\sim}S_+\}. \tag{16.12}$$

Combining (16.11) and (16.12), we have

$$H_P\times^{-1}S_+.$$

16.1.3.3 Double antecedents weakly mutually inverse intersection connections transform to single antecedent ones

$$\{P\cap Q|\cap^{-1}R\}\cap\{P\cap Q|\cap^{-1}{\sim}R\}. \tag{16.13}$$

Formula (16.13) is called double antecedent weakly mutually inverse intersection connection, because there are two antecedents P and Q. According to (16.14), double antecedents weakly mutually inverse intersection connections can be transformed into single antecedent ones.

$$\{P\cap Q|\cap^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}\cap\{Q|\cap^{-1}R\}. \tag{16.14}$$

Observing the rows of examinees 4 and 5 in Table 16.1, we can establish

$$\{E_B\cap H_F|\cap^{-1}S_+\}\cap\{E_B\cap H_F|\cap^{-1}{\sim}S_+\}. \tag{16.15}$$

From $E_B\cap H_F|\cap^{-1}S_+$ in (16.15) and (16.14), using second-level hypothetical inference, we can infer

$$\{E_B|\cap^{-1}S_+\}\cap\{H_F|\cap^{-1}S_+\}. \tag{16.16}$$

From $E_B\cap H_F|\cap^{-1}{\sim}S_+$ in (16.15) and (16.14), using second-level hypothetical inference, we can infer

$$\{E_B|\cap^{-1}{\sim}S_+\}\cap\{H_F|\cap^{-1}{\sim}S_+\}. \tag{16.17}$$

Combining (16.16) and (16.17), we have

$$\{E_B|\cap^{-1}S_+\}\cap\{E_B|\cap^{-1}{\sim}S_+\}, \tag{16.18}$$

and

$$\{H_F|\cap^{-1}S_+\}\cap\{H_F|\cap^{-1}{\sim}S_+\}. \tag{16.19}$$

Thus, we transform the double antecedents weakly mutually inverse intersection connection (16.15) into two single antecedent ones (16.18) and (16.19).

16.1.3.4 Chain reasoning of mutually inverse intersection connections

According to (16.20), we can make chain reasoning on $|\cap^{-1}$.

$$\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}. \tag{16.20}$$

For example, from examinee 1 in Table 16.1, we have

$$H_P|\cap^{-1}E_A. \tag{16.21}$$

Take the intersection of (16.21) and (16.1) as the minor premise (see (16.22)),

$$\{H_P|\cap^{-1}E_A\}\cap\{E_A\subseteq^{-1}S_+\}, \tag{16.22}$$

Take (16.20) as the major premise, use the second-level hypothetical inference, we can infer

$$H_P|\cap^{-1}S_+. \tag{16.23}$$

16.1.3.5 Conclusions on mutually-inversistic rough set

Conventional rough set can only mine decision rules; i.e., mutually inverse being contained connections. Mutually-inversistic rough set, in addition to mutually inverse being contained connections, can also mine mutually inverse intersection connections, strongly mutually inverse intersection connections, weakly mutually inverse intersections, it can also make chain reasoning.

16.2 Weakly mutually inverse intersection connection mining algorithm

16.2.1 Weakly mutually inverse intersection connection mining algorithm

Suppose the database to be mined has m attributes. On each attribute an equivalence relation is defined. The database has n tuples.

Weakly mutually inverse intersection connection mining algorithm is divided into the upper level and the lower level. The upper level is attribute couple determination algorithm, the lower level is weakly mutually inverse intersection connection determination algorithm.

The attribute couple determination algorithm:

Suppose *i* and *j* range over the attributes.

For i=1 to m−1 step 1

For *j*=*i*+1 to m step 1

{Select *i* and *j* as the attribute couple to be mined;}

Termination of the upper level algorithm.

The weakly mutually inverse intersection determination algorithm is divided into three steps:

Step 1:

(Suppose the attribute couple the upper algorithm selects is attributes e and f.)

For e=1 to n step 1

{Change the attribute value to its corresponding partition block, say, E_p;}

For f=1 to n step 1

{Change the attribute value to its corresponding partition block, say, F_q;}

Step 2:

For e=1 to n step 1

For f=1 to n step 1

(Suppose e is E_p, f is F_q)

{Generate mutually inverse intersection connection $E_p|\cap^{-1}F_q$; /* if E_p and F_q are in the same tuple, then they have mutually inverse intersection connection. */

If the bank of mutually inverse intersection connections does not have $E_p|\cap^{-1}F_q$, then insert it into the bank;} /* Up to now, all mutually inverse intersection connections have been mined.*/

Step 3:

(Suppose the bank of mutually inverse intersection connections have r mutually inverse intersection connections, numbered 1, 2, ..., r.)

For k=1 to r−1 step 1

(Suppose the kth mutually inverse intersection connection is $E_k|\cap^{-1}F_k$;)

For l=k+1 to r step 1

{If the lth mutually inverse intersection connection is $E_k|\cap^{-1}F_l$, and $F_l \neq F_k$,

then generate weakly mutually inverse intersection connection $\{E_k|\cap^{-1}F_k\}\cap\{E_k|\cap^{-1}\sim F_k\}$;

if $\{E_k|\cap^{-1}F_k\}\cap\{E_k|\cap^{-1}\sim F_k\}$ is not in the bank of weakly mutually inverse intersection connections,

then insert it into the bank;

else if the lth mutually inverse intersection connection is $E_l|\cap^{-1}F_k$, and $E_l \neq E_k$,

then generate weakly mutually inverse intersection connection $\{F_k|\cap^{-1}E_k\}\cap\{F_k|\cap^{-1}\sim E_k\}$;

if $\{F_k|\cap^{-1}E_k\}\cap\{F_k|\cap^{-1}\sim E_k\}$ is not in the bank of weakly mutually inverse intersection connections,

then insert it into the bank;}

Termination of the lower level algorithm.

16.2.2 An example

Example 16.2: Suppose the knowledgebase is the first five examinees of Table 16.1, redrawn as Table 16.2.

Table 16.2 College entrance exam knowledgebase

U	E	H	S
1	A	P	+
2	A	F	+
3	B	P	+
4	B	F	+
5	B	F	−

Now, we exert weakly mutually inverse intersection connection mining algorithm on Table 16.2. Suppose after the upper level algorithm is exerted, the attributes selected are i=1 and j=3; i.e., E and S.

Now, we exert the lower level algorithm on E and S.

After step 1 is exerted, Table 16.2 becomes Table 16.3.

Table 16.3 Exert step 1 on Table 16.2

E	S
E_A	S_+
E_A	S_+
E_B	S_+
E_B	S_+
E_B	S_-

In Table 16.3, E_A denotes the partition block for score A, E_B for score B, S_+ denotes the partition block for admission to colleges, S_- for not admission to college.

Exerting step 2 to Table 16.3, we obtain the bank of mutually inverse intersection connections $\{E_A|\cap^{-1}S_+, E_B|\cap^{-1}S_+, E_B|\cap^{-1}S_-\}$.

Exerting step 3 on the bank of mutually inverse intersection connections, we obtain the bank of weakly mutually inverse intersection connections $\{\{S_+|\cap^{-1}E_A\}\cap\{S_+|\cap^{-1}\sim E_A\}, \{E_B|\cap^{-1}S_+\}\cap\{E_B|\cap^{-1}\sim S_+\}\}$.

16.3 Mutually-inversistic fuzzy rough set

Mutually-inversistic fuzzy rough set is the integration of mutually-inversistic rough set and fuzzy rough set.

16.3.1 Fuzzy rough set

Fuzzy rough set is defined as follows.

Suppose K=(U, S) is a Pawlak approximation space; i.e., R∈ind(K) is an equivalence relation on the universe U, $\underset{\sim}{A}$ is a fuzzy set on U, $\forall x\in$U, then we define that the lower and upper approximations of the fuzzy set $\underset{\sim}{A}$ with respect to knowledge R are $\underline{R\underset{\sim}{A}}$ and $\overline{R\underset{\sim}{A}}$ respectively, they are all fuzzy subsets on U, their membership functions are:

$\mu\underline{R\underset{\sim}{A}}(x)=\inf\{\mu_{\underset{\sim}{A}}(y)|\forall y\in[x]_R\}$,

$\mu\overline{R\underset{\sim}{A}}(x)=\sup\{\mu_{\underset{\sim}{A}}(y)|\forall y\in[x]_R\}$.

16.3.2 Mutually-inversistic fuzzy rough set

Example 16.3: Suppose we have a college entrance exam knowledgebase shown in Table 16.4.

Table 16.4 College entrance exam knowledgebase

U	Exam score E	Exam score type ET	Health checkup H	University admitted S
x_1	700	A	healthy	Tsinghua University
x_2	650	A	unhealthy	Peking University
x_3	600	B	healthy	BUAA
x_4	550	B	unhealthy	BUPT
x_5	500	C	healthy	BUIC
x_6	450	C	unhealthy	BUIS
x_7	400	D	healthy	BUU
x_8	350	D	unhealthy	BCC

ET is an equivalence relation, it divides the universe into four blocks: ET_A, ET_B, ET_C, ET_D. H is also an equivalence relation, it divides the universe into two blocks: $H_{healthy}$, and $H_{unhealthy}$. E is a quantitative attribute, if we want to convert it into a fuzzy set "high score" (denoted as $\underset{\sim}{E}$), then we need a membership function for it. Suppose the membership function is

$$\mu_{\underset{\sim}{E}}(E_i)=(E_i-350)/400,$$

where E_i is the exam score of x_i. For example, the exam score of x_1 is 700, converting to membership grade, it is (700−350)/400=0.875. Suppose the membership grade of Tsinghua University, Peking University, BUAA, BUPT, BUIC, BUIS, BUU, and BCC to "good university" ("good university" is a fuzzy set, denoted as $\underset{\sim}{S}$) are 1, 1, 0.9, 0.8, 0.4, 0.3, 0, and 0.

We change E and all of its attribute values into $\underset{\sim}{E}$, followed by their membership grades. We change S and all of its attribute values into $\underset{\sim}{S}$, followed by their membership grades, obtaining Table 16.5.

Table 16.5 Changes made to Table 16.4

U	$\underset{\sim}{E}$	$\mu_{\underset{\sim}{E}}$	ET	H	$\underset{\sim}{S}$	$\mu_{\underset{\sim}{S}}$
x_1	$\underset{\sim}{E}$	0.875	A	healthy	$\underset{\sim}{S}$	1
x_2	$\underset{\sim}{E}$	0.75	A	unhealthy	$\underset{\sim}{S}$	1
x_3	$\underset{\sim}{E}$	0.625	B	healthy	$\underset{\sim}{S}$	0.9
x_4	$\underset{\sim}{E}$	0.5	B	unhealthy	$\underset{\sim}{S}$	0.8
x_5	$\underset{\sim}{E}$	0.375	C	healthy	$\underset{\sim}{S}$	0.4
x_6	$\underset{\sim}{E}$	0.25	C	unhealthy	$\underset{\sim}{S}$	0.3
x_7	$\underset{\sim}{E}$	0.125	D	healthy	$\underset{\sim}{S}$	0
x_8	$\underset{\sim}{E}$	0	D	unhealthy	$\underset{\sim}{S}$	0

Exerting knowledge reduction algorithm of conventional rough set on Table 16.5, we can mine the mutually inverse being contained connections:

$$ET_A \subseteq^{-1} \underset{\sim}{S} \tag{16.24}$$

$$ET_B \subseteq^{-1} \underset{\sim}{S} \tag{16.25}$$

$$ET_C \subseteq^{-1} \underset{\sim}{S} \tag{16.26}$$

$$ET_D \subseteq^{-1} \underset{\sim}{S} \tag{16.27}$$

Considering their membership grades, (16.24) to (16.27) becomes semi-fuzzy mutually inverse being contained connections:

$$ET_A \subseteq^{-1} \underset{\sim}{S}\ [1] \tag{16.28}$$

$$ET_B \subseteq^{-1} \underset{\sim}{S}\ [0.9] \tag{16.29}$$

$$ET_B \subseteq^{-1} \underset{\sim}{S}\ [0.8] \tag{16.30}$$

$$ET_C \subseteq^{-1} \underset{\sim}{S}\ [0.4] \tag{16.31}$$

$$ET_C \subseteq^{-1} \underset{\sim}{S}\ [0.3] \tag{16.32}$$

$$ET_D \subseteq^{-1} \underset{\sim}{S}\ [0] \tag{16.33}$$

For example, 0.4 in (16.31) is the upper approximation of C with respect to $\underset{\sim}{S}$ under the partition ET $\mu\overline{ET}\underset{\sim}{S}(C)$, 0.3 in (16.32) is the lower approximation $\mu\underline{ET}\underset{\sim}{S}(C)$.

Taking (16.28) to (16.33) respectively as the minor premise, taking

$$\{P \subseteq^{-1} Q\} \subseteq^{-1} \{P | \cap^{-1} Q\} \tag{16.34}$$

$$\{P \subseteq^{-1} Q\} =^{-1} \sim \{P | \cap^{-1} \sim Q\} \tag{16.35}$$

as the major premise, using second-level hypothetical inference as the inference rule, we can infer semi-fuzzily strongly mutually inverse intersection connections. For example, from (16.31), (16.34), and (16.35) we can infer

$$\{ET_C | \cap^{-1} \underset{\sim}{S}\}[0.4] \cap \sim \{ET_C | \cap^{-1} \sim \underset{\sim}{S}\}[0.4] \tag{16.36}$$

Formula (16.36) can be transformed into semi-fuzzily weakly mutually inverse intersection connection

$$\{ET_C | \cap^{-1} \underset{\sim}{S}\}[0.4] \cap \{ET_C | \cap^{-1} \sim \underset{\sim}{S}\}[0.6] \tag{16.37}$$

where $0.6 = \mu\underline{ET} \sim \underset{\sim}{S}(C) = 1 - \mu\overline{ET}\underset{\sim}{S}(C) = 1 - 0.4$. Likewise, we can infer

$$\{H_{healthy} | \cap^{-1} \underset{\sim}{E}\}[0.875] \cap \{H_{healthy} | \cap^{-1} \sim \underset{\sim}{E}\}[0.125] \tag{16.38}$$

which says: the upper approximation of healthy examinees having high scores is 0.875, the lower approximation of healthy examinees not having high scores is 0.125.

The third and seventh columns of Table 16.5 are fuzzy sets, we can mine them using the fuzzy association rule mining algorithm shown in Fig. 15.7, mining

$$\underset{\sim}{E} \subseteq^{-1} \underset{\sim}{S}\ [0.86] \tag{16.39}$$

which says: having higher score is bound to be admitted by a better university.

In mutually-inversistic set theory, there is set theorem

$$\{P | \cap^{-1} Q\} \cap \{Q \subseteq^{-1} R\} \subseteq^{-1} \{P | \cap^{-1} R\} \tag{16.40}$$

which can be used for chain reasoning on $|\cap^{-1}$. Taking the intersection of $\{H_{healthy} | \cap^{-1} \underset{\sim}{E}\}$

[0.875] in (16.38) and (16.39) as the minor premise (see (16.41)),

$$\{H_{healthy}|\cap^{-1}\underset{\sim}{E}\}[0.875]\cap\{\underset{\sim}{E}\subseteq^{-1}\underset{\sim}{S}\}[0.86] \tag{16.41}$$

taking (16.40) as the major premise, using second-level hypothetical inference as the inference rule, we can infer

$$H_{healthy}|\cap^{-1}\underset{\sim}{S}\,[0.75] \tag{16.42}$$

where 0.75=0.875*0.86, it is the upper approximation of (16.42), which says: healthy examinees can be admitted into good university.

16.4 Mutually-inversistic mathematical morphology

Mutually-inversistic mathematical morphology is the integration of conventional morphology and mutually-inversistic rough set.

16.4.1 Mathematical morphology

In conventional mathematical morphology, the dilation of A by B is defined as

$A\oplus B=\{z|(B^{\wedge})_z\cap A\neq\varnothing\}$.

The erosion of A by B is defined as

$A\Theta B=\{z|(B)_z\subseteq A\}$.

In mutually-inversistic mathematical morphology, the dilation of A by B is defined as

$A\oplus^{-1}B=\{z|(B^{\wedge})_z|\cap^{-1}A\}$.

The erosion of A by B is defined as

$A\Theta^{-1}B=\{z|(B)_z\subseteq^{-1}A\}$.

And

$$\sim\{A\Theta^{-1}B\}=^{-1}\sim A\oplus^{-1}B^{\wedge} \tag{16.43}$$

holds.

Proof (16.43): $\sim\{A\Theta^{-1}B\}=^{-1}\{z|\sim\{(B)_z\subseteq^{-1}A\}\}=^{-1}\{z|(B)_z|\cap^{-1}\sim A\}=^{-1}\sim A\oplus^{-1}B^{\wedge}$.

16.4.2 Boundary extraction algorithm

The conventional boundary extraction algorithm is

$A-(A\Theta B)$.

That is the boundary of a set A is obtained by first eroding A by B and then performing the set difference between A and its erosion. Suppose a binary image is shown in Fig. 16.2, the structuring element is shown in Fig. 16.3, then $A\Theta B$ is shown in Fig. 16.4, $A-(A\Theta B)$ is shown in Fig. 16.5, from which we see that the boundary is the outer edge of the binary image.

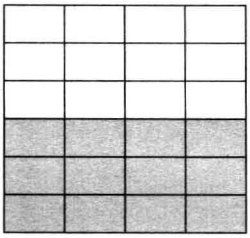


Fig. 16.2 Binary image A

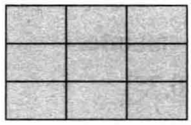


Fig. 16.3 The structuring element

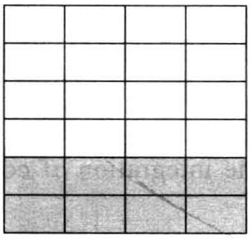


Fig. 16.4 AΘB

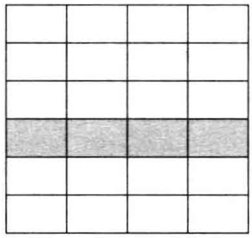


Fig. 16.5 A−(AΘB)

The mutually-inversistic mathematical morphology based boundary extraction algorithm is

$$\{A\oplus^{-1}B\}\cap\{\sim A\oplus^{-1}B\}.$$

Suppose the binary image is shown in Fig. 16.2, the structuring element is shown in Fig. 16.3, then $A\oplus^{-1}B$ is shown in Fig. 16.6, $\sim A\oplus^{-1}B$ is shown in Fig. 16.7, $\{A\oplus^{-1}B\}\cap\{\sim A\oplus^{-1}B\}$ is shown in Fig. 16.8, from which we see that the boundary is the outer edge of the binary image and the inner edge of the background, doubling the thickness of the boundary of the conventional boundary extraction algorithm.

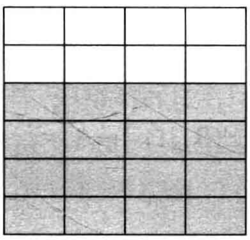


Fig. 16.6 $A\oplus^{-1}B$

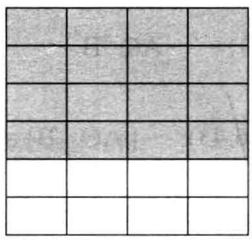


Fig. 16.7 $\sim A\oplus^{-1}B$

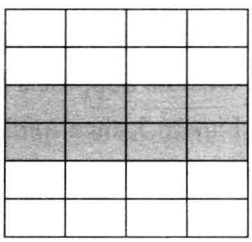


Fig. 16.8 $\{A\oplus^{-1}B\}\cap\{\sim A\oplus^{-1}B\}$

16.5 Mutually-inversistic evidence theory

In conventional evidence theory, belief function is defined as

$\mathrm{Bel}(X)=\sum_{A\subseteq X}m(A) \qquad \forall X\in\rho(U).$

Likelihood function is defined as

$\mathrm{Pl}(X)=\sum_{A\cap X\neq\varnothing}m(A) \qquad \forall X\in\rho(U).$

In mutually-inversistic evidence theory, belief function is defined as

$\mathrm{Bel}(X)=\sum_{A\subseteq^{-1}X}m(A) \qquad \forall X\in\rho(U).$

Likelihood function is defined as

$\mathrm{Pl}(X)=\sum_{A|\cap^{-1}X}m(A) \qquad \forall X\in\rho(U).$

16.6 Mutually-inversistic point set topology

In conventional point set topology, suppose X is a nonempty set, T is the topology of X. Suppose $A\subset(X, T)$, $x\in X$. If $\forall u\in U$, we have $u\cap(A/\{x\})\neq\varnothing$, then x is called the limit point of A. $A^0=\{x|\exists u\in U, x\in u\subset A\}$ is called the inner core of A. If $x\in X$ and $\forall u\in U_x$, we have $u\cap A\neq\varnothing$ and $u\cap A^c\neq\varnothing$, then x is called the boundary point of A.

In mutually-inversistic point set topology, suppose X is a nonempty set, T is the topology of X. Suppose $A\subset(X, T)$, $x\in X$. If $\forall u\in U$, we have $u|\cap^{-1}(A/\{x\})$, then x is called the limit point of A. $A^0=\{x|\exists u\in U, x\in u\subseteq^{-1}A\}$ is called the inner core of A. If $\forall u\in U_x$, we have $\{u|\cap^{-1}A\}\cap\{u|\cap^{-1}A^c\}$, then x is called the boundary point of A.

16.7 Mutually-inversistic concept lattice

In conventional concept lattice, for a formal concept (U, V, R), we define a pair of set-theoretic operators $*$, $\#$: $2^U\rightarrow2^V$, called the sufficiency and dual sufficiency operators, as follows:

$X^*=\{y\in V|X\subseteq Ry\},$

$X^\#=\{y\in V|X^c\cap(Ry)^c\neq\varnothing\}.$

In mutually-inversistic concept lattice, we define the sufficiency and dual sufficiency operators as follows:

$X^*=\{y\in V|X\subseteq^{-1}Ry\},$

$X^\#=\{y\in V|X^c|\cap^{-1}(Ry)^c\}.$

16.8 Mutually-inversistic second-type covering rough set

In conventional covering rough set, suppose set $X \subseteq U$, $C_*(X)=\{K \in C | K \subseteq X\}$ is called the covering lower approximation set family. In conventional second-type covering rough set, $\overline{X}=\cup\{K \in C | K \cap X \neq \varnothing\}$ is called the second-type covering upper approximation.

In mutually-inversistic covering rough set, suppose set $X \subseteq U$, $C_*(X)=\{K \in C | K \subseteq^{-1} X\}$ is called the covering lower approximation set family. In mutually-inversistic second-type covering rough set, $\overline{X}=\cup\{K \in C | K | \cap^{-1} X\}$ is called the second-type covering upper approximation.

Part 6
Unified logics

Mutually-inversistic logic is unified logics. It unifies classical logic, Aristotelian logic, ancient Chinese logic, mutually-inversistic modal logic, relevance logic, inductive logic, many-valued logic, Boolean algebra, fuzzy logic, natural deduction, paraconsistent logic, non-monotonic logic. It is the unifications of mathematical logic and philosophical logic, of ancient logic and modern logic, of western logic and Chinese logic, of two-valued logic and many-valued logic, of inductive logic and deductive logic, of crisp logic and fuzzy logic, of extensional logic and intensional logic.

Chapter 17
Unified logics

17.1 Unification of classical logic

Unified logics unify classical logic. Classical logic has two calculi and four theories. Unified logics also have two calculi and four theories. Unified logics inherit Frege's principle from classical logic. The truth tables for $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ are the same as Tables 1.6, 1.8, 1.9, 2.4, and 2.7 respectively. In classical logic, individual variables are first-order variables, function variables and predicate variables are second-order variables. In unified logics, term variables are first-order variables, function variables and predicate variables are second-order variables. Invariable propositions and variable propositions in unified logics correspond to propositions and proposition functions in classical logic.

17.2 Unification of Aristotelian logic

Single concept, singular judgment, universal judgment, and existential judgment in Aristotelian logic correspond to term, fact proposition, single empirical or mathematical connection proposition with the outmost connection operator being $\leqslant^{-1}$, and single empirical or mathematical connection proposition with the outmost connection operator being $/\wedge^{-1}$. Inferences in Aristotelian logic, including immediate inference and syllogism, are inference rules. If they are transformed into theorems, then they are the single logical theorems in unified logics. There are five Euler diagrams in Aristotelian logic, corresponding to Figs. 2.7, 2.8, 2.9, 2.10, and 2.12 respectively.

17.3 Unification of ancient Chinese logic

2400 years ago, Chinese philosopher Mozi defined "dagu" (A is a sufficient condition of B) with "youzhibiran" (if A is true, then B is necessarily true). That is, in ancient Chinese logic, necessary connection is the same as sufficient conditional relationship, which is inherited by unified logics.

17.4 Unification of mutually-inversistic modal logic

Mutually-inversistic modal logic has two features: (1) necessary connection is the same

as sufficient conditional relationship. (2) there are four modalities: necessity, probability, reality, possibility, they are all binary ones, and necessity vs. probability, reality vs. possibility are two categories of dialectic logic.

If man(x)$\leqslant^{-1}$mortal(x) is established by implicit inductive composition, then we have "it is necessary that man be mortal", denoted by man(x)$\leqslant_{i}^{-1}$mortal(x). If man(x)$\leqslant^{-1}$mortal(x) is established by explicit inductive composition, then we have "it is probable that man be mortal", denoted by man(x)$\leqslant_{e}^{-1}$mortal(x). If man(x)$/\wedge^{-1}$mortal(x) is established by explicit inductive composition, then we have "it is realistic that man be mortal", denoted by man(x)$/\wedge_{e}^{-1}$mortal(x). If man(x)$/\wedge^{-1}$mortal(x) is established by implicit inductive composition, then we have "it is possible that man be mortal", denoted by man(x)$/\wedge_{i}^{-1}$mortal(x). The four propositions form a chain shown in Fig. 17.1.

man(x)$\leqslant_{i}^{-1}$mortal(x)

man(x)$\leqslant_{e}^{-1}$mortal(x)

man(x)$/\wedge_{e}^{-1}$mortal(x)

man(x)$/\wedge_{i}^{-1}$mortal(x)

Fig. 17.1 Chain of mutually-inversistic modal logic

In the chain of Fig. 17.1, the upper proposition mutually inversely implies the lower proposition.

Mutually-inversistic modal logic also has a logical square shown in Fig. 17.2.

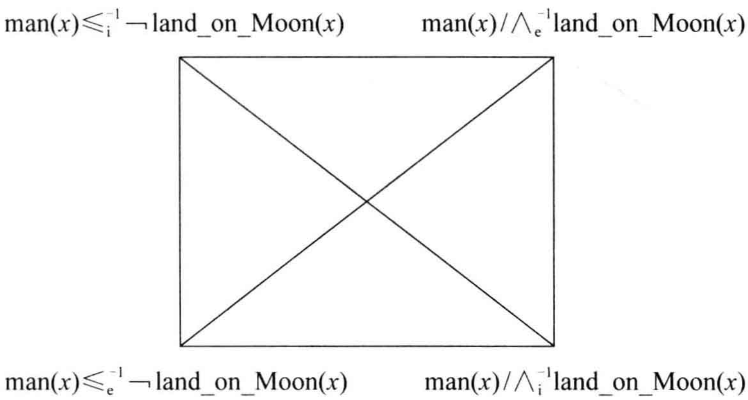


Fig. 17.2 Logical square of mutually-inversistic modal logic

Fig. 17.2 tells us that in the history of mankind, man(x)$\leqslant_{e}^{-1}\neg$land_on_Moon(x) was being truthified, but we had not been able to truthify man(x)$\leqslant_{i}^{-1}\neg$land_on_Moon(x). Contrarily, at the beginning of the twentieth century, scientists had found out that as long as the speed

of a spaceship reaches the second cosmic velocity, it can escape the attraction of the Earth and fly to the Moon. This is to say, scientists had truthified man(x)$/\wedge_i^{-1}$land_on_Moon(x). At this time, both man(x)$\leqslant_e^{-1}\neg$land_on_Moon(x) and man(x)$/\wedge_i^{-1}$land_on_Moon(x) had been true. Through man's continuous effort, at last, on July 20, 1969, Neil Armstrong landed on the Moon. Possibility turned to reality. We truthified man(x)$/\wedge_e^{-1}$land_on_Moon(x), falsifying man(x)$\leqslant_e^{-1}\neg$land_on_Moon(x) at the same time.

17.5 Unification of extensional logic and intensional logic

Explicit inductive composition concerns extension. Implicit inductive composition concerns intension.

17.6 Unification of relevance logic

In relevance logic, A relevantly implying B, if A and B share the same variable. In unified logics, a variable is relevantly bound or quasi-relevantly bound, if it is shared by both the antecedent and the consequent.

17.7 Unification of inductive logic and deductive logic

From Fig. 2.4, we learn that first-level explicit inductive composition belongs to induction, first-level implicit inductive composition and first-level decomposition belong to deduction.

17.8 Unification of two-valued logic and many-valued logic

Superficially, unified logics have four values: F, T, n, and u. Since n and u cannot be generated, actually, they have only two values: F and T.

17.9 Unification of Boolean algebra

In unified logics, logical theorems are proved by main-auxiliary algebras of set theorems (see Section 23.3), which are variants of Boolean algebra.

17.10 Unification of fuzzy logic

Unified logics unifies fuzzy logic, see Chapter 15.

17.11 Unification of natural deduction

In unified logics, logical theorems are proved by second-level implicit inductive composition without logical axioms.

17.12 Unification of paraconsistent logic

Paraconsistent logic allows contradictions. Because $P \wedge \neg P \rightarrow Q$ does not hold, the contradictions will not spread. In unified logics, $P \wedge \neg P \leqslant^{-1} Q$ also does not hold. It is a quasi-logical connection proposition with the outmost connection operator being $\leqslant^{-1}$, the antecedent is meaningless, Q is free, inductive composition cannot go on.

17.13 Unification of non-monotonic logic

In non-monotonic logic, when evidence increases, conclusion decreases, at least not increase; i.e., $(P \rightarrow R) \rightarrow (P \wedge Q \rightarrow R)$ does not hold. In unified logics, $\{P \leqslant^{-1} R\} \leqslant^{-1} \{P \wedge Q \leqslant^{-1} R\}$ also does not hold. It is a single logical connection proposition, Q is free, inductive composition cannot go on.

Part 7
Mutually-inversistic analytic geometry

Mutually-inversistic analytic geometry is the expansion of conventional analytic geometry. Points on the Cartesian coordinate axes are continuous. While points on the coordinate axes of mutually-inversistic analytic geometry can be continuous or discrete, can be terms or facts. Mutually-inversistic analytic geometry consists of analytic geometry of terms and analytic geometry of facts, which are based on coordinate system of terms and coordinate system of facts respectively

Chapter 18
Mutually-inversistic analytic geometry

18.1 Analytic geometry of terms

18.1.1 Linear analytic geometry of terms

In a linear coordinate system of terms, there is only one axis: the x-axis.

Example 18.1: Truthify the first-order single empirical or mathematical connection proposition nat(x)$\subseteq'^{-1}$int(x).

Solution: This example is one of mathematical abstraction. Set nat(x) is composed of the vertices in Fig. 18.1 marked with "○", set int(x) is composed of the vertices in Fig. 18.1 marked with "×". From Fig. 18.1, we see that nat(x) is a subset of int(x), therefore the proposition holds.

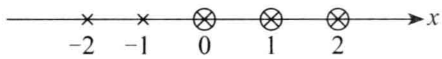


Fig. 18.1 truthification of nat(x)$\subseteq'^{-1}$int(x)

18.1.2 Planar analytic geometry of terms

In a planar coordinate system of terms, there are two axes: the x-axis and y-axis.

Example 18.2: Truthify parent(x, y)$\subseteq'^{-1}$ancestor(x, y).

Solution: This example is one of empirical abstraction. Suppose John is the parent of Bob, Bob is the parent of Sam. Set parent(x, y) is composed of the vertices in Fig. 18.2 marked with "△", set ancestor(x, y) is composed of the vertices in Fig. 18.2 marked with "□". From Fig. 18.2, we see that parent(x, y) is a subset of ancestor(x, y), therefore the proposition holds.

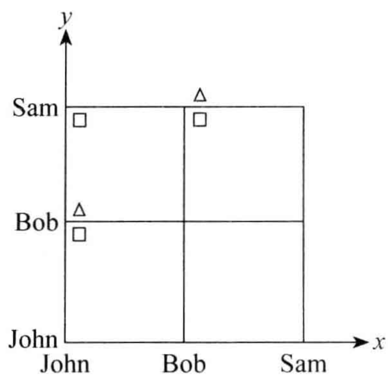


Fig. 18.2 Truthification of parent(x, y)$\subseteq'^{-1}$ancestor(x, y)

18.1.3 Spatial analytic geometry of terms

In a spatial coordinate system of terms, there are three axes: the x-axis, y-axis, and z-axis.

18.1.3.1 $Xr_1y \cap yr_2z \subseteq^{-1} xr_3z$ type propositions

In $xr_1y \cap yr_2z \subseteq^{-1} xr_3z$ type propositions, r_1, r_2, and r_3 are binary relations of empirical abstraction such as parent, ancestor, or mathematical abstraction such as=, $<$. This type of propositions needs the concept of projection cylinder of spatial curves, so, let us review it first.

In conventional analytic geometry, suppose we have two equations of curved surface:

$$F_1(x, y, z)=0, \tag{18.1}$$

$$F_2(x, y, z)=0. \tag{18.2}$$

The intersection of the two equations is a spatial curve:

$$\begin{cases} F_1(x, y, z)=0, \\ F_2(x, y, z)=0. \end{cases} \tag{18.3}$$

If y is eliminated from (18.3), then the projection cylinder of the spatial curve is obtained:

$$F(x, z)=0. \tag{18.4}$$

Example 18.3: Suppose the two equations of spherical surfaces are:

$$x^2+y^2+z^2=1, \tag{18.5}$$

$$x^2+(y-1)^2+(z-1)^2=1. \tag{18.6}$$

Calculate their projection cylinder on the y axis.

Solution: Subtract (18.5) from (18.6), and reduce, we obtain:

$$y+z=1. \tag{18.7}$$

Substitute $y=1-z$ into (18.5), we then obtain the projection cylinder:

$$x^2+2z^2-2z=0. \tag{18.8}$$

The establishment of $xr_1y \cap yr_2z \subseteq^{-1} xr_3z$ type propositions is as follows: xr_1y corresponds to (18.1), yr_2z to (18.2), $xr_1y \cap yr_2z$ to (18.3), the projection cylinder of $xr_1y \cap yr_2z$ to (18.4), if the projection cylinder of $xr_1y \cap yr_2z$ is mutually inversely contained in xr_3z, then $xr_1y \cap yr_2z \subseteq^{-1} xr_3z$ is truthified, otherwise, it is falsified.

Example 18.4: Truthify $x=y \cap y=z=^{-1} x=z$.

Solution: For the straight lines (1, 1, z), (2, 2, z), and (3, 3, z) in Fig. 18.3 (marked with "△"), $x=y$ hold. For the straight lines (x, 1, 1), (x, 2, 2), and (x, 3, 3) in Fig. 18.3 (marked with "□") $y=z$ hold. $X=y \cap y=z$, the intersection of $x=y$ and $y=z$, is the vertices that mark with both "△" and "□", that is, vertices A, B, and C in Fig. 18.3. Making projection cylinder along the y-axis through A, B, and C, the projection cylinder is the same as $x=z$, which is the straight lines (1, y, 1), (2, y, 2), and (3, y, 3) in Fig. 18.3 (marked with "*"). Therefore, $x=y \cap y=z=^{-1} x=z$ is truthified.

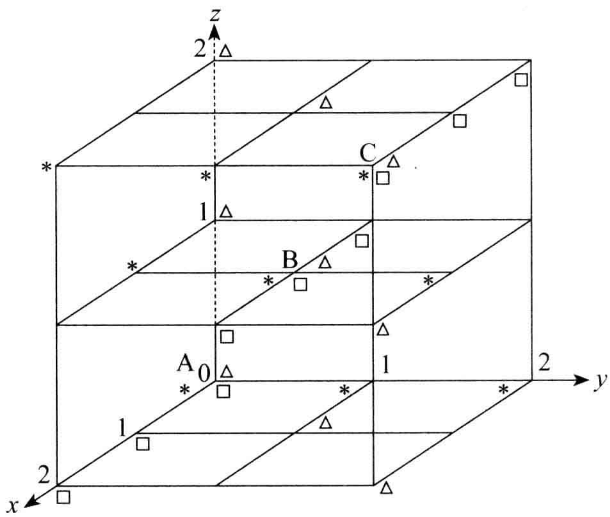


Fig. 18.3 truthification of $x=y \cap y=z=^{-1}x=z$

Example 18.5: Falsity parent(x, y) $\cap$ parent(y, z) $\subseteq^{-1}$ parent(x, z).

Solution: Suppose John is the parent of Bob, and Bob is the parent of Sam. From Fig.18.4, we learn that, parent(x, y)is composed of the two straight lines (John, Bob, z) and (Bob, Sam, z) marked with "△", parent(y, z) is composed of the two straight lines (x, John, Bob) and (x, Bob, Sam) marked with "□". The intersection of parent(x, y) and parent(y, z) is the vertex A that is marked with both "△" and "□". Making the projection cylinder along the y-axis through vertex A, we obtain the straight line (John, y, Sam). But, parent(x, z) is composed of the two straight lines (John, y, Bob) and (Bob, y, Sam). The projection cylinder is not mutually inversely contained in parent(x, z). Therefore, parent(x, y) $\cap$ parent(y, z) $\subseteq^{-1}$ parent(x, z) is falsified.

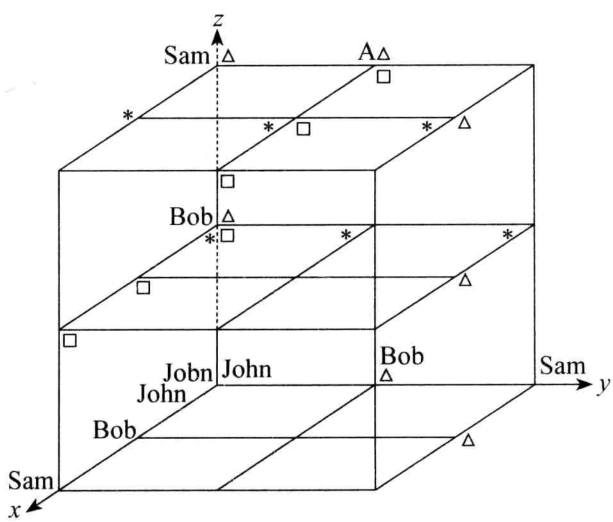


 Fig. 18.4 Falsification of parent(x, y) $\cap$ parent(y, z) $\subseteq^{-1}$ parent(x, z)

18.1.3.2 Non-$xr_1y \cap yr_2z \subseteq^{-1} xr_3z$ type propositions

Example 18.6: Truthify $x<y \cap x<z \subseteq^{-1} x<y+z$.

Solution: From Fig. 18.5, we learn that, $x<y$ is the straight line (0, 1, z) marked with "△", $x<z$ is the straight line (0, y, 1) marked with "□". $X<y \cap x<z$ is vertex A that is marked with both "△" and "□".$X<y+z$ is the vertices marked with "*". Vertex A is mutually inversely contained in $x<y+z$. Therefore, $x<y \cap x<z \subseteq^{-1} x<y+z$ is truthified.

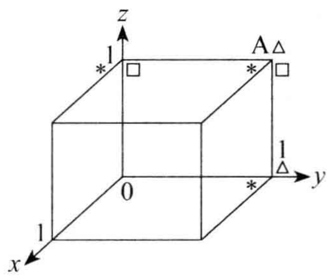


Fig. 18.5 Truthification of $x<y \cap x<z \subseteq^{-1} x<y+z$

18.2 Analytic geometry of facts

18.2.1 Planar analytic geometry of facts

In a planar coordinate system of facts, there are two coordinate axes of facts: the P-axis and Q-axis.

Example 18.7: Truthify the second-order single set connection proposition $\{P \subseteq^{-1} Q\} \subseteq^{-1} \{P | \cap^{-1} Q\}$.

Solution: We use the mathematical abstraction. In Fig. 18.6, $P \subseteq^{-1} Q$ is composed of the vertices marked with "△", $P | \cap^{-1} Q$ is composed of the vertices marked with "□". From Fig. 18.6, we see that $P \subseteq^{-1} Q$ is mutually inversely contained in $P | \cap^{-1} Q$. Therefore, $\{P \subseteq^{-1} Q\} \subseteq^{-1} \{P | \cap^{-1} Q\}$ is truthified.

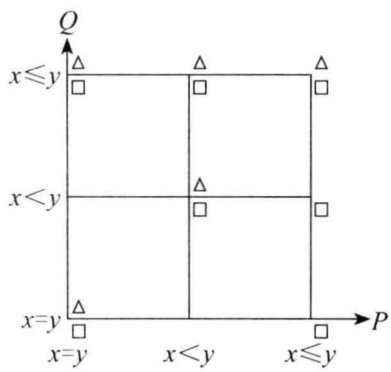


Fig. 18.6 Truthification of $\{P \subseteq^{-1} Q\} \subseteq^{-1} \{P | \cap^{-1} Q\}$

Example 18.8: Truthify $\{P{\subset^{-1}}Q\}{\subseteq^{-1}}\{P{\subseteq^{-1}}Q\}$.

Solution: We use the set-theoretic abstraction. (The points on the coordinate axes of facts cannot be empty sets.) In Fig. 18.7, $P{\subset^{-1}}Q$ is composed of the vertices marked with "△", $P{\subseteq^{-1}}Q$ is composed of the vertices marked with "□". From Fig. 18.7, we see that $P{\subset^{-1}}Q$ is mutually inversely contained in $P{\subseteq^{-1}}Q$. Therefore, $\{P{\subset^{-1}}Q\}{\subseteq^{-1}}\{P{\subseteq^{-1}}Q\}$ is truthified.

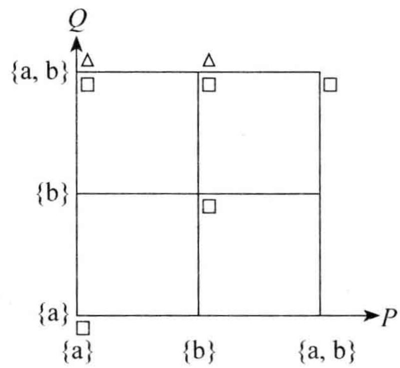


Fig. 18.7 Truthification of $\{P{\subset^{-1}}Q\}{\subseteq^{-1}}\{P{\subseteq^{-1}}Q\}$

18.2.2 Spatial analytic geometry of facts

In a spatial coordinate system of facts, there are three coordinate axes of facts: P-, Q-, and R-axes.

18.2.2.1 $P\varphi_1Q\cap Q\varphi_2R{\subseteq^{-1}}P\varphi_3R$ type propositions

In $P\varphi_1Q\cap Q\varphi_2R{\subseteq^{-1}}P\varphi_3R$ type propositions, φ_1, φ_2, and φ_3 are empirical or mathematical connection operators, such as $=^{-1}$, $\subset^{-1}$, $\subseteq^{-1}$, and $|\cap^{-1}$. These type of propositions can be established by empirical, mathematical, or set-theoretic abstractions. They will use the concept of projection cylinder of spatial curve. Their establishments are the lifting of $xr_1y\cap yr_2z{\subseteq^{-1}}xr_3z$ one level up.

Example 18.9: Truthify $\{P|{\cap^{-1}}Q\}\cap\{Q{\subseteq^{-1}}R\}{\subseteq^{-1}}\{P|{\cap^{-1}}R\}$.

Solution: We use the empirical abstraction. In Fig. 18.8, $P|{\cap^{-1}}Q$ is composed of the seven straight lines marked with "△": (parent(x, y), parent(x, y), R), (grandparent(x, y), grandparent(x, y), R), (ancestor(x, y), ancestor(x, y), R), (parent(x, y), ancestor(x, y), R), (ancestor(x, y), parent(x, y), R), (grandparent(x, y), ancestor(x, y), R), and (ancestor(x, y), grandparent(x, y), R). $Q{\subseteq^{-1}}R$ is composed of the five straight lines marked with "□": (P, parent(x, y), parent(x, y)), (P, grandparent(x, y), grandparent(x, y)), (P, ancestor(x, y), ancestor(x, y)), (P, parent(x, y), ancestor(x, y)), and (P, grandparent(x, y), ancestor(x, y)). $\{P|{\cap^{-1}}Q\}\cap\{Q{\subseteq^{-1}}R\}$, the intersection of $P|{\cap^{-1}}Q$ and $Q{\subseteq^{-1}}R$, is composed of the vertices marked with both "△" and "□", that is, vertices: A, B, C, D, E, F, G, H, I, J, K, through which we make projection cylinder along the Q-axis: (parent(x, y), Q, parent(x, y)),

(grandparent(x, y), Q, grandparent(x, y)), (ancestor(x, y), Q, ancestor(x, y)), (parent(x, y), Q, ancestor(x, y)), (ancestor(x, y), Q, parent(x, y)), (grandparent(x, y), Q, ancestor(x, y)), and (ancestor(x, y), Q, grandparent(x, y)). The projection cylinder coincides with the seven straight lines of $P|\cap^{-1}R$ extending along the Q-axis marked with "*". Therefore, $\{P|\cap^{-1} Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$ is truthified.

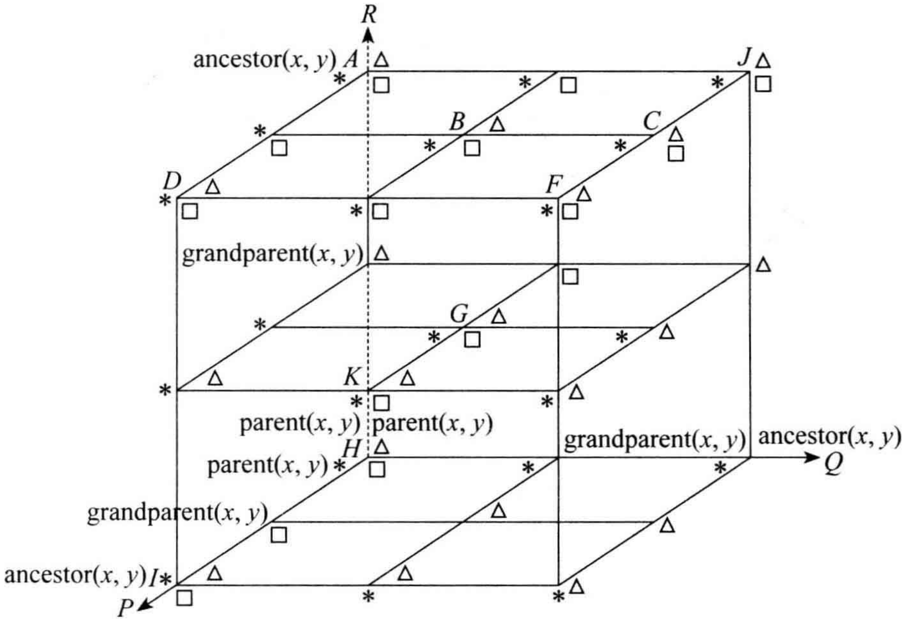


Fig. 18.8 Truthification of $\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$

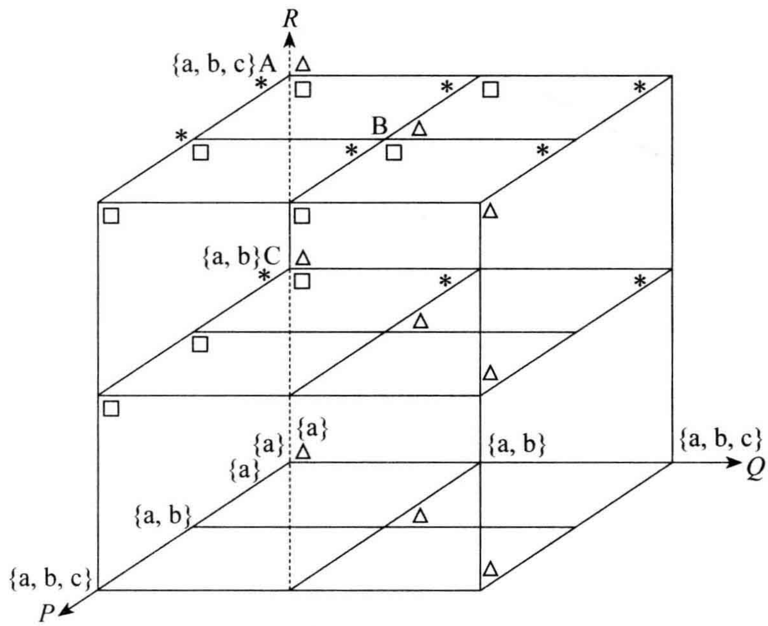


Fig. 18.9 truthification of $\{P=^{-1}Q\}\cap\{Q\subset^{-1}R\}\subseteq^{-1}\{P\subset^{-1}R\}$

Example 18.10: Truthify $\{P=^{-1}Q\}\cap\{Q\subset^{-1}R\}\subseteq^{-1}\{P\subset^{-1}R\}$.

Solution: We use the set-theoretic abstraction. In Fig. 18.9, $P=^{-1}Q$ is composed of the three straight lines marked with "△": ({a}, {a}, R), ({a, b}, {a, b}, R), and ({a, b, c}, {a, b, c}, R). $Q\subset^{-1}R$ is composed of the three straight lines marked with "□": (P, {a}, {a, b}), (P, {a, b}, {a, b, c}), and (P, {a}, {a, b, c}). $\{P=^{-1}Q\}\cap\{Q\subset^{-1}R\}$, the intersection of $P=^{-1}Q$ and $Q\subset^{-1}R$, is composed of the vertices marked with both "△" and "□", that is, vertices A, B, and C, through which we make projection cylinder along the Q-axis: ({a}, Q, {a, b, c}), ({a, b}, Q, {a, b, c}), and({a}, Q, {a, b}). The projection cylinder coincides with the three straight lines of $P\subset^{-1}R$ extending along the Q-axis marked with "∗". Therefore, $\{P=^{-1}Q\}\cap\{Q\subset^{-1}R\}\subseteq^{-1}\{P\subset^{-1}R\}$ is truthified.

Example 18.11: Falsify $\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$.

Solution: We use the mathematical abstraction. In Fig. 18.10, $P\subseteq^{-1}Q$ is composed of the five straight lines marked with "△": ($x=y$, $x=y$, R), ($x=y$, $x\leqslant y$, R), ($x<y$, $x<y$, R), ($x<y$, $x\leqslant y$, R), and ($x\leqslant y$, $x\leqslant y$, R). $Q|\cap^{-1}R$ is composed of the seven straight lines marked with "□": (P, $x=y$, $x=y$), (P, $x\leqslant y$, $x=y$), (P, $x<y$, $x<y$), (P, $x\leqslant y$, $x<y$), (P, $x=y$, $x\leqslant y$), (P, $x<y$, $x\leqslant y$), and (P, $x\leqslant y$, $x\leqslant y$). $\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}$, the intersection of $P\subseteq^{-1}Q$ and $Q|\cap^{-1}R$, are the vertices marked with both "△" and "□", through which we can make projection cylinder along the Q-axis. Two of the projection straight lines: ($x=y$, Q, $x<y$) and ($x<y$, Q, $x=y$) made through vertices A and B are not mutually inversely contained in $P|\cap^{-1}R$, which is composed of the seven straight lines extending along the Q-axis marked with "∗". Therefore, $\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$ is falsified.

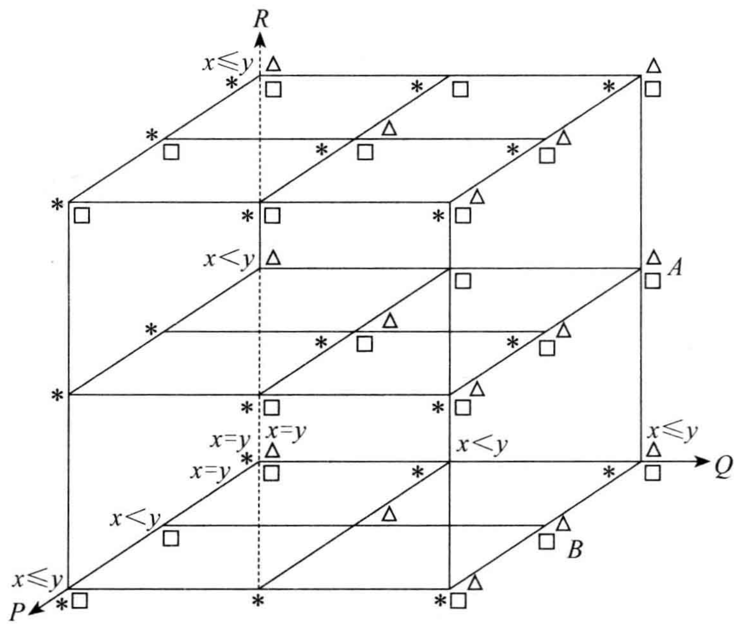


Fig. 18.10 Falsification of $\{P\subseteq^{-1}Q\}\cap\{Q|\cap^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$

18.2.2.2 Non-$P\varphi_1Q\cap Q\varphi_2R\subseteq^{-1}P\varphi_3R$ type propositions

Example 18.12: Truthify $\{P\subseteq^{-1}R\}\cap\{Q\subseteq^{-1}R\}=^{-1}\{P\cup Q\subseteq^{-1}R\}$.

Solution: We use the set-theoretic abstraction. In Fig. 18.11, $P\subseteq^{-1}R$ is composed of the five straight lines marked with "△": ({a}, Q, {a}), ({b}, Q, {b}), ({a, b}, Q, {a, b}), ({a}, Q, {a, b}), and ({b}, Q, {a, b}). $Q\subseteq^{-1}R$ is composed of the five straight lines marked with "□": (P, {a}, {a}), (P, {b}, {b}), (P, {a, b}, {a, b}), (P, {a}, {a, b}), and (P, {b}, {a, b}). $\{P\subseteq^{-1}R\}\cap\{Q\subseteq^{-1}R\}$ is composed of the vertices marked with both "△" and "□". $\{P\cup Q\subseteq^{-1}R\}$ is composed of the vertices marked with "*". From Fig. 18.11, we learn that a vertex marked with both "△" and "□" is bound to be marked with "*", vice versa. Therefore, $\{P\subseteq^{-1}R\}\cap\{Q\subseteq^{-1}R\}=^{-1}\{P\cup Q\subseteq^{-1}R\}$ is truthified.

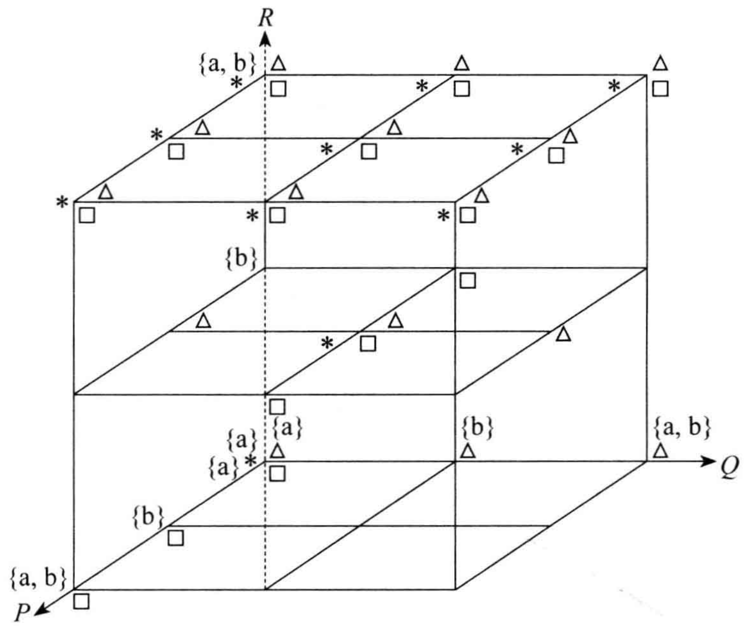


Fig. 18.11 Truthification of $\{P\subseteq^{-1}R\}\cap\{Q\subseteq^{-1}R\}=^{-1}\{P\cup Q\subseteq^{-1}R\}$

18.3 Loci of moving points

Establishments of propositions using mutually-inversistic analytic geometry are with the loci of moving points. Take Example 18.12 as an example. When the moving points move along the axes, they either reach a vertex not marked with both "△" and "□", or they reach one also marked with "*". Therefore, the proposition is truthified.

Part 8
Mutually-inversistic mathematical analysis

Numerical analysis deems that numerical differential and numerical integral are mutually inverse operations. Whereas mutually-inversistic mathematical analysis deems that difference quotient (part of numerical differential, regarded as discrete derivative) and the variant of summation formulas (regarded as discrete integral) are mutually inverse operations. Mutually-inversistic mathematical analysis consists of double-sided discrete calculus, single-sided discrete calculus, and unified calculus. Double-sided discrete calculus is based on the coordinate system of terms.

Chapter 19
Double-sided discrete calculus

Double-sidedness in derivative refers to that both the left derivative and the right one exist and equal to each other, in integral refers to that both the negative axis and the positive one are integrable.

19. 1 Double-sided discrete calculus of unary functions

19.1.1 Planar coordinate system of terms and double-sided unary functions

Double-sided unary function is in the form of $m=f(n)$, where n and m are discrete. $M=f(n)$ can be expressed by planar coordinate system of terms, points on the axis of which are numbers. The distance between numbers is h, which does not approach 0. Fig. 19.1 is an example of planar coordinate system of terms, on which $m=f(n)$ is denoted as marking the ordered pairs $(n, f(n))$ with "Δ". For example, $m=-2n$ is denoted by marking the ordered pairs in Fig. 19.1 with "Δ".

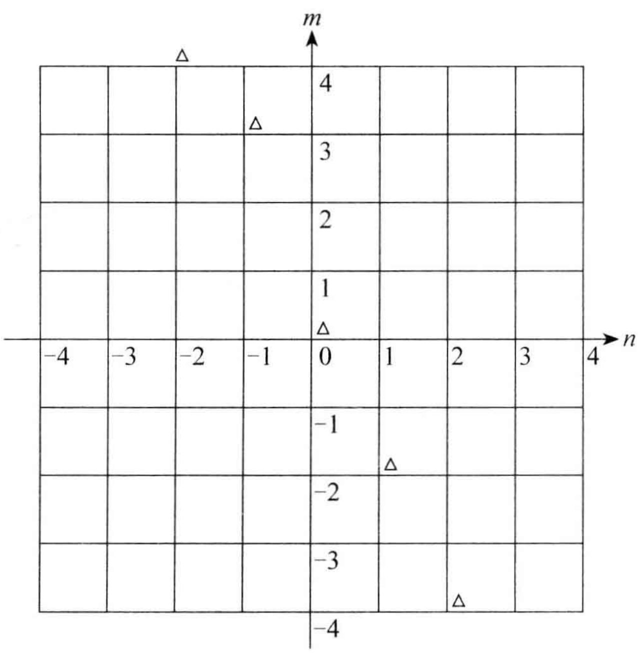


Fig. 19.1 Planar coordinate system of terms and double-sided unary functions

19.1.2 Derivatives

Definition 19.1: The right derivative of double-sided function $m=f(n)$ at n_0 is defined as the forward difference quotient

$$\frac{\Delta m}{\Delta n}\Big|_{n=n0}=\frac{f(n1)-f(n0)}{n1-n0}, \tag{19.1}$$

where n_1 is adjacent to n_0 at the right of the n axis.

The geometric interpretation of right derivative is as follows: Suppose we have a function m=f(n) denoted by the vertices marked with "Δ" in Fig. 19.2, in which the coordinates of P_0 are (n_0, f(n_0)), those of P_1 are (n_1, f(n_1)). We draw a straight line through P_0 and P_1, called the right tangent line of P_0, the slope

$$\frac{f(n1)-f(n0)}{n1-n0}$$

of which is the right derivative of m=f(n) at n_0.

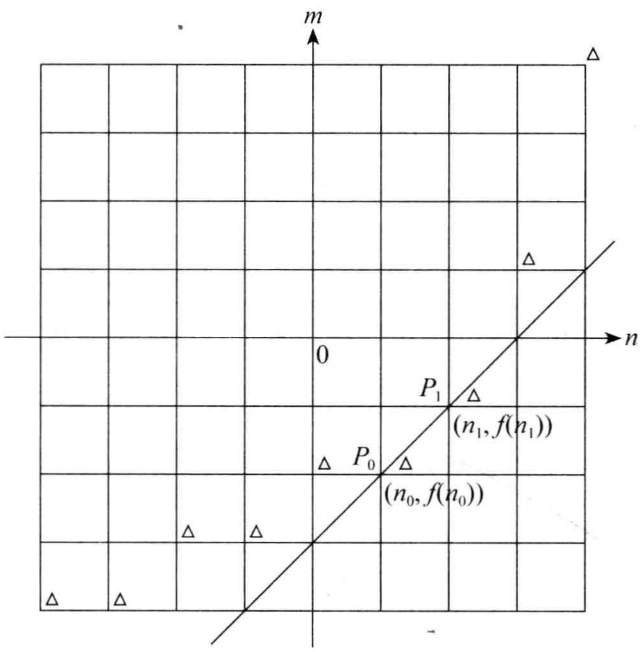


Fig.19.2 geometric interpretation of right derivative

Definition 19.2: the left derivative of $m=f(n)$ at n_0 is defined as the backward difference quotient

$$\frac{\nabla m}{\nabla n}\Big|_{n=n0}=(f(n_{-1})-f(n_0))/(n_{-1}-n_0), \tag{19.2}$$

where n_{-1} is adjacent to n_0 at the left of the n axis.

Similarly, the geometric interpretation of the left derivative of $m=f(n)$ at n_0 is the slope of the left tangent line through the point (n_0, f(n_0)).

Definition 19.3: The derivative of $m=f(n)$ at n_0 (denoted as $f'(n_0)$, $\frac{df}{dn}\Big|_{n=n0}$, $\frac{dm}{dn}\Big|_{n=n0}$) exists, if the left derivative equals the right derivative; i.e.,

$$\frac{\nabla m}{\nabla n}\Big|_{n=n0}=\frac{\Delta m}{\Delta n}\Big|_{n=n0}, \tag{19.3}$$

and we say that $m=f(n)$ is derivable at n_0.

Theorem 19.1: Function $m=f(n)$ is derivable at $n=n_0$, if and only if the three points $(n_{-1}, f(n_{-1}))$, $(n_0, f(n_0))$, $(n_1, f(n_1))$ are on the same straight line.

Proof: The derivability of $m=f(n)$ is equivalent to the holding of (19.3), which is equivalent to that the slope of the left tangent line equals the slope of the right one, which is equivalent to the coincidence of the left tangent line and the right one, which is equivalent to that the three points $(n_{-1}, f(n_{-1}))$, $(n_0, f(n_0))$, $(n_1, f(n_1))$ are on the same straight line.

Q.E.D.

Example 19.1: Suppose function $m=f(n)$ is denoted by the points marked with "Δ" in Fig. 19.3. Verify that the function is derivable at n_0.

Solution: The right derivative of function $m=f(n)$ at n_0 is $\frac{\Delta m}{\Delta n}\Big|_{n=n0}=\frac{f(n1)-f(n0)}{n1-n0}=2$, the left derivative is $\frac{\nabla m}{\nabla n}\Big|_{n=n0}=(f(n_{-1})-f(n_0))/(n_{-1}-n_0)=2$. The right derivative equals the left one. Therefore, $m=f(n)$ is derivable at n_0.

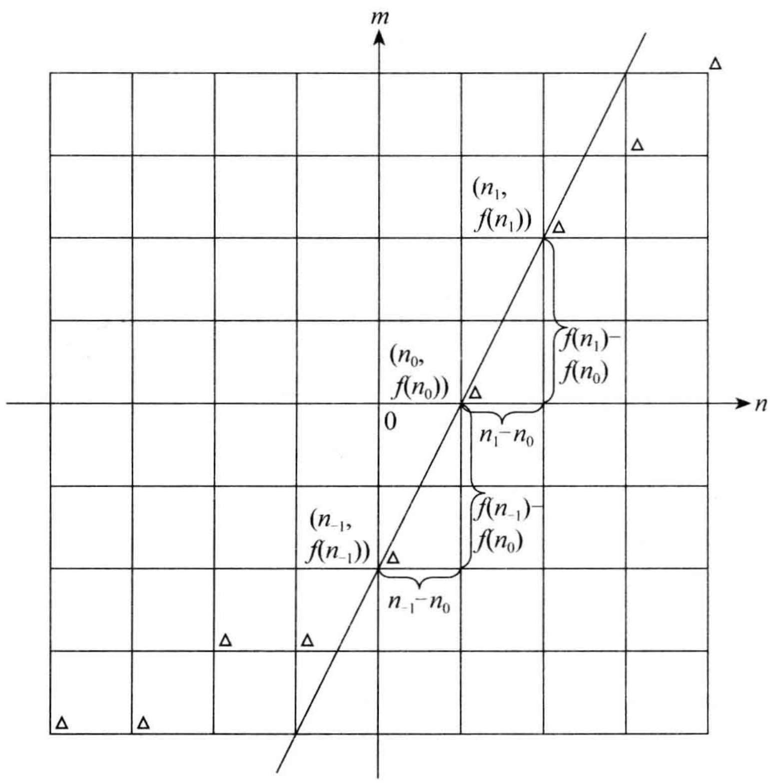


Fig. 19.3 Derivability of m=f(n) at n_0

19.1.3 Derivative functions

Definition 19.4: The derivative of function $m=f(n)$ is called derivative function, denoted as $f'(n)$, $\frac{dm}{dn}$, m', if for any n_i in the domain, $f'(n_i)$ exists.

Theorem 19.2: Function $m=f(n)$ has derivative function $f'(n)$, if and only if $f(n)$ is a straight line.

Proof: Necessity: Suppose function $m=f(n)$ has derivative function $f'(n)$, then $f'(n_i)$ exists. From Theorem 19.1, we know that the three points $(n_{i-1}, f(n_{i-1}))$, $(n_i, f(n_i))$, $(n_{i+1}, f(n_{i+1}))$ are on the same straight line. Because also $f'(n_{i+1})$ exists, the three points $(n_i, f(n_i))$, $(n_{i+1}, f(n_{i+1}))$, $(n_{i+2}, f(n_{i+2}))$ are on the same straight line. Because both $f'(n_i)$ and $f'(n_{i+1})$ exist, the four points $(n_{i-1}, f(n_{i-1}))$, $(n_i, f(n_i))$, $(n_{i+1}, f(n_{i+1}))$, and $(n_{i+2}, f(n_{i+2}))$ are on the same straight line. Repeat the above procedures, we can obtain that all points of $f(n)$ are on the same straight line.

Sufficiency: Suppose $f(n)$ is a straight line, then the three points $(n_{i-1}, f(n_{i-1}))$, $(n_i, f(n_i))$, $(n_{i+1}, f(n_{i+1}))$ are on the same straight line. According to Theorem 19:1, $f'(n_i)$ exists. Considering the fact that n_i is an arbitrary point in the n axis, therefore $f'(n)$ exists.

Q.E.D.

Suppose the domain and range of $m=f(n)$ are **Z** (integer domain), then only functions in the form of $m=pn+q$ ($p, q\in\mathbf{Z}$) has derivative functions, because only they are the straight lines. For example, suppose $m=-3n+2$, then $m'=-3$. Suppose $m=4$, then $m'=0$. If no confusion occurs, then we simply call derivative functions derivatives.

19.1.4 Derivatives of compound functions

Definition 19.5: Suppose $l=f(m)$ and $m=g(n)$, then, similar to Newtonian calculus, we have $\frac{dl}{dn}=\frac{dl}{dm}*\frac{dm}{dn}=f'(m)*g'(n)$.

Example 19.2: Suppose $l=-4m$, $m=3n$, compute $\frac{dl}{dn}$.

Solution: $\frac{dl}{dm}=-4$, $\frac{dm}{dn}=3$, hence, $\frac{dl}{dn}=\frac{dl}{dm}*\frac{dm}{dn}=(-4)*3=-12$.

19.1.5 Higher-order derivatives

Definition 19.6: Suppose $m=pn+q$, then $\frac{dm}{dn}=p$. The second-order derivative of m to n is: $d^2m/dn^2=\frac{d}{dn}\left(\frac{dm}{dn}\right)=0$.

19.1.6 Integrals

First, let us see an example of summation formula: $\sum_{n=-2}^{2}3=15$. In this example, n ranges

over −2, −1, 0, 1, and 2 five values. For each value of *n*, a 3 is added to the partial sum, and five additions in total, hence the sum is 15. Double-sided discrete integral is similar to this example. The only difference is that the first value of *n* is not included. *N* ranges over −1, 0, 1, and 2 four values. For each value of *n*, a 3 is added to the partial integral, and four additions in total, hence the integral is 12. The integral can be denoted as $3n|_{-2}^{2}=3*(2-(-2)=3*4=12$. In order to distinguish the double-sided discrete integral with the summation formula, we denoted the integral as:

$$(\nabla n=1)\mathrm{I}_{n=-2}^{2}3=3n|_{-2}^{2}, \tag{19.4}$$

where I is the integral sign, $\nabla n=n_i-n_{i-1}$ is the integral step; i.e., the distance between two adjacent points on the *n* axis, *n* is the integral variable, 3 is the integrand, −2 is the lower limit of the integral, 2 is the upper limit of the integral, 3*n* is the primitive function.

Integral step can be any constant. Suppose ∇n is 0.5, then the above integral becomes

$$(\nabla n=0.5)\mathrm{I}_{n=-2}^{2}3=\frac{3n}{0.5}|_{-2}^{2}=6*(2-(-2))=6*4=24.$$

Because every time *n* is assigned a value, a 3 is added to the partial integral, and *n* is assigned 1/0.5=2 values in an unit length, and there are 2−(−2)=4 unit length, thus, the integral is 24.

Generally, we have

Definition 19.7: Suppose $f(n)$=p, p is a constant, then the double-sided discrete integral of $f(n)$ is

$$(\nabla n=h)\mathrm{I}_{n=a}^{b}p=\frac{pn}{h}|_{a}^{b}=\frac{p}{h}(b-a). \tag{19.5}$$

The geometric interpretation of double-sided discrete integral is the sum of the length of the vertical lines enclosed by *m*=p, *n*=a+h, *n*=b, and the *n* axis, shown in Fig. 19.4.

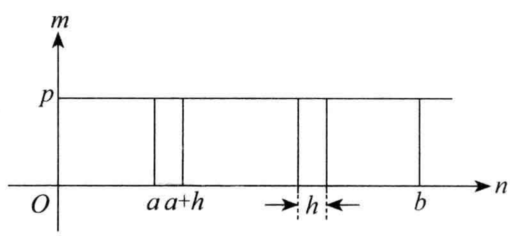


Fig. 19.4 Geometric interpretation of double-sided discrete integral

The geometric interpretation of Newtonian integral is the area of curved trapezoid, it is the infinite sum to the differential of area. The geometric interpretation of double-sided discrete integral is not the area of curved trapezoid, no need for the differential of area. Therefore, the concept of differential is not introduced into double-sided discrete calculus.

19.2 Double-sided discrete calculus of binary functions

19.2.1 Spatial coordinate system of terms and double-sided discrete binary functions

A double-sided discrete binary function is in the form of $l=f(m, n)$, which can be expressed by the spatial coordinate system of terms, an example of which is shown in Fig. 19.5.

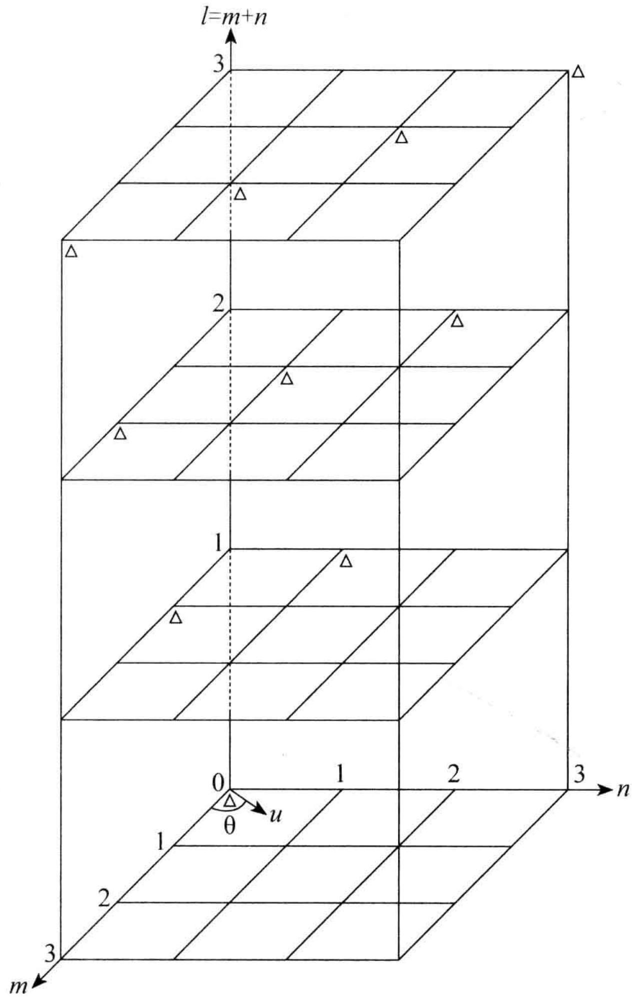


Fig. 19.5 Spatial coordinate system of terms and double-sided discrete binary functions

$L=f(m, n)$ is denoted in the coordinate system of terms by marking the triple tuples $(m, n, f(m, n))$ with "Δ". For example, $l=m+n$ is denoted by the vertices in Fig. 19.5 marked with "Δ".

19.2.2 Partial derivatives

Suppose $l=f(m, n)$ is a function of two variables m and n. If n is held constant, say $n=n_0$, then $l=f(m, n_0)$ is a function of the single variable m. Its right derivative at $m=m_0$ is called the right partial derivative of f with respect to m at (m_0, n_0) and is denoted by $\frac{\Delta l}{\Delta m}(m, n)|_{m=m0,\ n=n0}$ or $\frac{\Delta f}{\Delta m}(m, n)|_{m=m0,\ n=n0}$.

Definition 19.8: $\frac{\Delta l}{\Delta m}(m, n)|_{m=m0,\ n=n0}$ is defined as

$$\frac{\Delta l}{\Delta m}(m, n)|_{m=m0,\ n=n0}=\frac{f(m1, n0)-f(m0, n0)}{m1-m0},$$

where m_1 is the right neighbor of m_0.

Definition 19.9: The left partial derivative of $l=f(m, n)$ with respect to m at (m_0, n_0), denoted as $\frac{\nabla l}{\nabla m}(m, n)|_{m=m0,\ n=n0}$ or $\frac{\nabla f}{\nabla m}(m, n)|_{m=m0,\ n=n0}$, is defined as

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=m0,\ n=n0}=(f(m_{-1}, n_0)-f(m_0, n_0))/(m_{-1}-m_0),$$

where m_{-1} is the left neighbor of m_0.

Definition 19.10: The partial derivative of $l=f(m, n)$ with respect to m at (m_0, n_0), denoted as $\frac{\partial l}{\partial m}(m, n)|_{m=m0,\ n=n0}$ or $\frac{\partial f}{\partial m}(m, n)|_{m=m0,\ n=n0}$, exists, if the left partial derivative of $l=f(m, n)$ with respect to m at (m_0, n_0) equals the right one; i.e.,

$$\frac{\Delta l}{\Delta m}(m, n)|_{m=m0,\ n=n0}=\frac{\Delta l}{\Delta m}(m, n)|_{m=m0,\ n=n0}.$$

Theorem 19.3: $\frac{\partial l}{\partial m}(m, n)|_{m=m0,\ n=n0}$ exists, if and only if the three points $(m_{-1}, n_0, f(m_{-1}, n_0))$, $(m_0, n_0, f(m_0, n_0))$, and $(m_1, n_0, f(m_1, n_0))$ are on the same straight line.

Proof: Similar to the proof of Theorem 19.1, so omitted here.

Definition 19.11: If, when $n=n_0$, for any m_i, $\frac{\partial l}{\partial m}(m, n)|_{m=mi,\ n=n0}$ exists, then we say that the linear partial derivative function of f to m at $n=n_0$ exists, denoted by $\frac{\partial l}{\partial m}(m, n)|_{n=n0}$.

Theorem 19.4: $\frac{\partial l}{\partial m}(m, n)|_{n=n0}$ exists, if and only if $l=f(m, n_0)$ is a straight line.

Proof: Similar to the proof of Theorem 19.2, so omitted here.

$\frac{\Delta l}{\Delta n}(m, n)|_{m=m0,\ n=n0}$, $\frac{\nabla l}{\nabla n}(m, n)|_{m=m0,\ n=n0}$, $\frac{\partial l}{\partial n}(m, n)|_{m=m0,\ n=n0}$, and $\frac{\partial l}{\partial n}(m, n)|_{m=m0}$ can be similarly discussed.

Definition 19.12: If for any (m, n), the partial derivative of f to m exists, then we say that the surface partial derivative function of f to m exists, denoted by $\frac{\partial l}{\partial m}(m, n)$ or $\frac{\partial f}{\partial m}(m, n)$.

Similarly, we can define $\frac{\partial \mathrm{l}}{\partial \mathrm{n}}(\mathrm{m}, \mathrm{n})$. If no confusion occurs, then we simply call linear partial derivative function and surface partial derivative function partial derivative.

When computing $\frac{\partial \mathrm{l}}{\partial \mathrm{m}}(\mathrm{m}, \mathrm{n})$, we regard n as constant. Likewise, when computing $\frac{\partial \mathrm{l}}{\partial \mathrm{n}}(\mathrm{m}, \mathrm{n})$, we regard m as constant.

There are two kind of partially derivable functions: $l=\mathrm{p}m+\mathrm{q}n+\mathrm{r}$ and $l=\mathrm{p}mn+\mathrm{q}$.

Example 19.3: Suppose $l=m+n$ shown in Fig. 19.5, then

$$\frac{\partial \mathrm{l}}{\partial \mathrm{m}}(\mathrm{m}, \mathrm{n})|_{\mathrm{m}=0,\, \mathrm{n}=0}=\frac{\partial \mathrm{l}}{\partial \mathrm{m}}(\mathrm{m}, \mathrm{n})|_{\mathrm{n}=0}=\frac{\partial \mathrm{l}}{\partial \mathrm{m}}(\mathrm{m}, \mathrm{n})=1,$$

$$\frac{\partial \mathrm{l}}{\partial \mathrm{n}}(\mathrm{m}, \mathrm{n})|_{\mathrm{m}=0,\, \mathrm{n}=0}=\frac{\partial \mathrm{l}}{\partial \mathrm{n}}(\mathrm{m}, \mathrm{n})|_{\mathrm{m}=0}=\frac{\partial \mathrm{l}}{\partial \mathrm{n}}(\mathrm{m}, \mathrm{n})=1.$$

Example 19.4: Suppose $l=mn$ shown in Fig. 19.6, then

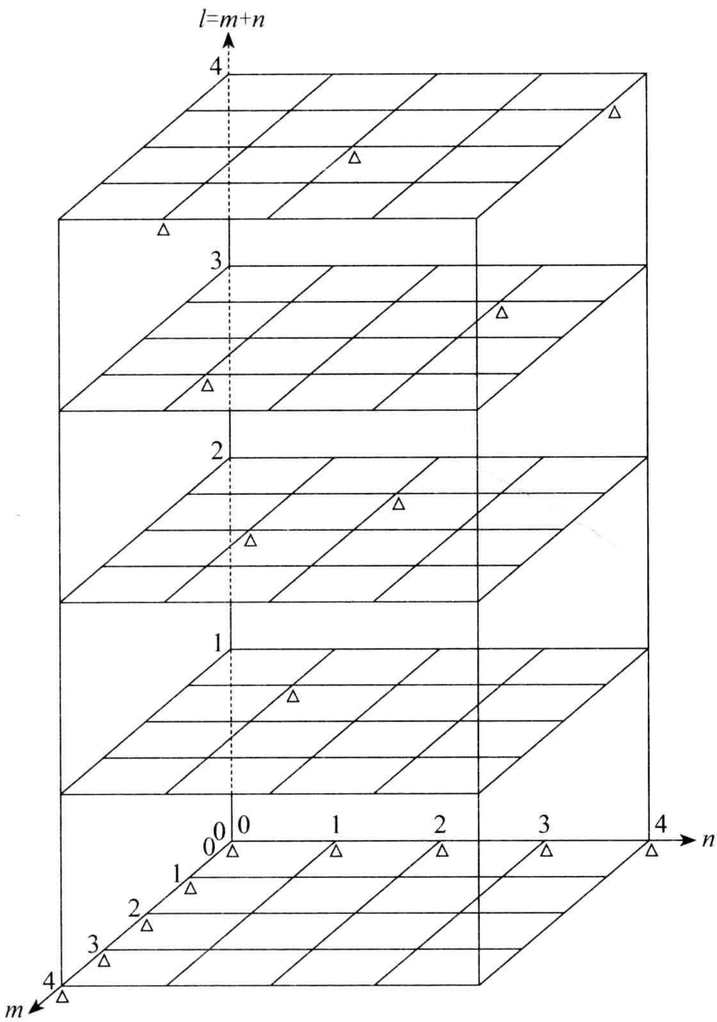


Fig. 19.6 Partial derivative for l=mn

$$\frac{\partial l}{\partial m}(m, n)|_{m=1, n=2}=\frac{\partial l}{\partial m}(m, n)|_{n=2}=2, \frac{\partial l}{\partial m}(m, n)=n,$$

$$\frac{\partial l}{\partial n}(m, n)|_{m=1, n=2}=\frac{\partial l}{\partial n}(m, n)|_{m=1}=1, \frac{\partial l}{\partial n}(m, n)=m.$$

The second-order partial derivative of f to m, denoted by $\partial^2 l/\partial m^2(m, n)$, is defined as $\frac{\partial}{\partial m}\left(\frac{\partial l}{\partial m}(m, n)\right)$. Similarly, $\partial^2 l/\partial n^2(m, n)$, $\partial^2 l/\partial m\partial n(m, n)$, and $\partial^2 l/\partial n\partial m(m, n)$ can be defined.

Example 19.5: Suppose $l=m+n$, then

$\partial^2 l/\partial m^2(m, n)=\partial^2 l/\partial m\partial n(m, n)=\partial^2 l/\partial n\partial m(m, n)=\partial^2 l/\partial n^2(m, n)=0.$

Example 19.6: Suppose $l=mn$, then

$\partial^2 l/\partial m^2(m, n)=\partial^2 l/\partial n^2(m, n)=0,$

$\partial^2 l/\partial m\partial n(m, n)=\partial^2 l/\partial n\partial m(m, n)=1.$

Definition 19.13: Directional derivative is defined as

$$\mathrm{Duf}(m, n)=\frac{\partial f}{\partial m}(m, n)\cos\theta+\frac{\partial f}{\partial n}(m, n)\sin\theta,$$

where u is the unit vector in the m-n plane, θ is the angle between u and positive axis of m, shown in Fig. 19.5.

Example 19.7: Suppose $l=m+n$, $\theta=\pi/4$, then

$\mathrm{Duf}(m, n)=1*\sqrt{2}/2+1*\sqrt{2}/2=\sqrt{2}.$

19.2.3 Double integrals

Definition 19.14: Double integral is defined as

$$(\nabla m=h_1)I_{m=a}{}^{b}(\nabla n=h_2)I_{n=c}{}^{d}p=(\nabla m=h_1)I_{m=a}{}^{b}p\left(\frac{d-c}{h2}\right)=p\left(\frac{d-c}{h2}\right)\left(\frac{b-a}{h1}\right).$$

Example 19.8: Compute $(\nabla m=0.2)I_{m=-3}{}^{1}(\nabla n=0.5)I_{n=0}{}^{5}7$.

Solution: $(\nabla m=0.2)I_{m=-3}{}^{1}(\nabla n=0.5)I_{n=0}{}^{5}7$

$$=(\nabla m=0.2)I_{m=-3}{}^{1}7*\left(\frac{5-0}{0.5}\right)$$

$$=7*\left(\frac{5-0}{0.5}\right)*\left(\frac{1-(-3)}{0.2}\right)$$

$$=7*10*20$$

$$=1400.$$

Chapter 20 Single-sided discrete calculus

In single-sided discrete calculus, backward difference quotient (regarded as single-sided left derivative) and the variants of summation formulas (regarded as single-sided discrete integral) are mutually inverse operations. Single-sidedness in derivative means that only the left derivative exists, in integral means that only the positive axis is integrable. Newtonian calculus defines derivative first, integral second. While single-sided discrete calculus develops from the known summation formulas, so it defines integral first, derivative second.

20.1 Single-sided discrete calculus of unary functions

20.1.1 Single-sided discrete definite integrals of variable upper limit and single-sided discrete infinite integrals

20.1.1.1 Concept of single-sided discrete definite integral of variable upper limit

Example 20.1: A power plant discharges 1 ton of smoke the 0^{th} day, 1/2 ton of smoke the first day, 1/4 ton of smoke the second day, …, $1/2^n$ ton of smoke the *n*th day. How many tons of smoke the power plant discharges from the 0^{th} day to the *n*th day?

Solution: According to the summation formula

$$\sum_{k=0}^{n} a^k=\{1-a^{n+1}\}/\{1-a\}, \tag{20.1}$$

where a=1/2, we know that from the 0^{th} day to the *n*th day, the power plant discharges

$$\sum_{k=0}^{n}\left(\frac{1}{2}\right)^k=\{1-(1/2)^{n+1}\}/\{1-(1/2)\} \tag{20.2}$$

tons of smoke.

Example 20.2: The problem is the same as Example 20.1. The question is: how many tons of smoke the power plant discharges from the first day to the *n*th day?

Solution: The smoke discharged the first day is the smoke discharged from the 0^{th} day to the first day 1+1/2 tons minus the smoke discharged the 0^{th} day 1 ton, that is

$$\{1-(1/2)^{1+1}\}/\{1-1/2\}-\{1-(1/2)^{0+1}\}/\{1-1/2\},$$

denoted as

$$(\nabla k=1)I_{k=0}^{1}(1/2)^k=\{1-(1/2)^{k+1}\}/\{1-1/2\}|_0^1.$$

The smoke discharged from the first day to the second day is the smoke discharged

from the 0^{th} day to the second day 1+1/2+1/4 tons minus the smoke discharged the 0^{th} day 1 ton, that is

$$\{1-(1/2)^{2+1}\}/\{1-1/2\}-\{1-(1/2)^{0+1}\}/\{1-1/2\},$$

denoted as

$$(\nabla k=1)I_{k=0}^{2}(1/2)^{k}=\{1-(1/2)^{k+1}\}/\{1-1/2\}|_{0}^{2}.$$

The smoke discharged from the first day to the *n*th day is the smoke discharged from the 0^{th} day to the *n*th day $1+1/2+1/4+\ldots+1/2^{n}$ minus the smoke discharged the 0^{th} day 1 ton, that is

$$\{1-(1/2)^{n+1}\}/\{1-1/2\}-\{1-(1/2)^{0+1}\}/\{1-1/2\},$$

denoted as

$$(\nabla k=1)I_{k=0}^{n}(1/2)^{k}=\{1-(1/2)^{k+1}\}/\{1-1/2\}|_{0}^{n}. \tag{20.3}$$

Formula (20.3) is a variant of (20.2). Formula (20.2) is used to compute the smoke discharged from the 0^{th} day to the *n*th day, while (20.3) is used to compute the smoke discharged from the 0^{th} day to the *n*th day minus the smoke discharged the 0^{th} day.

Formula (20.3) is a formula of single-sided discrete definite integrals of variable upper limit: I is the integral sign, k is the integral variable, 0 is the lower limit, n is the upper limit, ∇k is the integral step, representing the value the integral variable k increases each time, when $\nabla k=1$, k varies from 0, to 1, to 2, ..., $(1/2)^{k}$ is the integrand, $\{1-(1/2)^{k+1}\}/\{1-1/2\}$ is the primitive function.

Definition 20.1: $M=f(n)$ is a single-sided discrete function, if $n\in\mathbf{N}$ (natural number) and $m\in\mathbf{R}$ (real number).

Definition 20.2: Suppose $f(n)$ and $F(n)$ are single-sided discrete functions. If

$F(0)=f(0)$

$F(1)=f(0)+f(1)$

$F(2)=f(0)+f(1)+f(2)$

……

$F(n)=\sum_{k=0}^{n}f(k)$

Then we say that $F(n)$ is the sum of $f(k)$, and call

$\sum_{k=0}^{n}f(k)=F(n)$

a summation formula.

Definition 20.3: Suppose $f(n)$ and $F(n)$ are single-sided discrete functions. If

$F(0)=f(0)-f(0)$

$F(1)=f(0)+f(1)-f(0)$

$F(2)=f(0)+f(1)+f(2)-f(0)$

……

$F(n)=\sum_{k=0}^{n}f(k)-f(0)$

Then we say that $F(n)$ is the single-sided discrete definite integrals of variable upper limit

of $f(k)$, denoted by

$$(\nabla k=1)I_{k=0}{}^{n}f(k)=F(n), \tag{20.4}$$

where I is the integral sign, k is the integral variable, 0 is the lower limit, n is the upper limit, ∇k is the integral step, $f(k)$ is the integrand, $F(n)$ is the primitive function.

From Definitions 20.2 and 20.3 we know that formulas of single-sided discrete definite integrals of variable upper limit can be obtained by subtracting $f(0)$ from summation formulas; i.e.,

$$(\nabla k=1)I_{k=0}{}^{n}f(k)=\sum_{k=0}^{n}f(k)-f(0). \tag{20.5}$$

All summation formulas with the lower limit being 0, upper limit being n can be transformed into formulas of single-sided discrete definite integrals of variable upper limit like this.

20.1.1.2 Formulas of single-sided discrete definite integrals of variable upper limit

The following are some of the important formulas of single-sided discrete definite integrals of variable upper limit.

Integrals of exponential functions:

$$(\nabla k=1)I_{k=0}{}^{n}a^{k}=\{1-a^{n+1}\}/\{1-a\}-1 \quad (a<1) \tag{20.6}$$

$$(\nabla k=1)I_{k=0}{}^{n}a^{k}=\{a^{n+1}-1\}/\{a-1\}-1 \quad (a>1) \tag{20.7}$$

Integral of power functions:

$$(\nabla k=1)I_{k=0}{}^{n}1=n \tag{20.8}$$

$$(\nabla k=1)I_{k=0}{}^{n}k=n(n+1)/2=\{n^2+n\}/2 \tag{20.9}$$

$$(\nabla k=1)I_{k=0}{}^{n}k^2=n(n+1)(2n+1)/6=\{2n^3+3n^2+n\}/6 \tag{20.10}$$

$$(\nabla k=1)I_{k=0}{}^{n}k^3=n^2(n+1)^2/4=\{n^4+2n^3+n^2\}/4 \tag{20.11}$$

$$(\nabla k=1)I_{k=0}{}^{n}k^4=\{6n^5+15n^4+10n^3-n\}/30 \tag{20.12}$$

Integrals of products:

$$(\nabla k=1)I_{k=0}{}^{n}ka^{k}=\{[(a-1)n-1]a^{n+1}+a\}/(a-1)^2 \tag{20.13}$$

$$(\nabla k=1)I_{k=0}{}^{n}k^2a^{k}=\{[(a-1)n-1]^2a^{n+1}+a^{n+2}-a(a+1)\}/(a-1)^3 \tag{20.14}$$

Formulas (20.6) to (20.13) can be transformed from the known summation formulas using (20.5). Now, let us prove (20.14).

Proof: A in (20.13) is continuous, it can take any real number except 1. We take continuous derivative (the derivative in Newtonian calculus) on a to both sides of (20.13), obtaining

$$\begin{aligned}&(\nabla k=1)I_{k=0}{}^{n}k^2a^{k-1}\\&=\{[[(a-1)n-1]a^{n+1}+a]'(a-1)^2-[[(a-1)n-1]a^{n+1}+a][(a-1)^2]'\}/(a-1)^4\\&=\{n^2a^{n+2}-2n^2a^{n+1}-2na^{n+1}+a^{n+1}+n^2a^n+2na^n+a^n-a-1\}/(a-1)^3\end{aligned} \tag{20.15}$$

Both sides of (20.15) times a, and the following formula is obtained:

$(\nabla k=1)I_{k=0}^{n}k^2a^k$
$=\{n^2a^{n+3}-2n^2a^{n+2}-2na^{n+2}+n^2a^{n+1}+2na^{n+1}+a^{n+1}+a^{n+2}-a^2-a\}/(a-1)^3$
$=\{[(a-1)n-1]^2a^{n+1}+a^{n+2}-a(a+1)\}/(a-1)^3$

Q.E.D.

20.1.1.3 Single-sided discrete infinite integrals

Let the upper limit of the formulas of single-sided discrete definite integrals of variable upper limit approach infinity, then we obtain single-sided discrete infinite integrals. For example, when $a<1$,

$$(\nabla k=1)I_{k=0}^{\infty}a^k=I_{k=0}^{n}a^k=1-a^{n+1}\}/\{1-a\}-1=1/\{1-a\}-1 \tag{20.16}$$

$(\nabla k=1)I_{k=0}^{\infty}ka^k$
$=I_{k=0}^{n}ka^k$
$=[(a-1)n-1]a^{n+1}+a\}/(a-1)^2$
$$=a/(a-1)^2 \tag{20.17}$$

20.1.1.4 Properties of single-sided discrete definite integrals of variable upper limit

Single-sided discrete definite integrals of variable upper limit have the following properties.

Property 20.1: $(\nabla k=1)I_{k=0}^{n}(f(k)+g(k))= (\nabla k=1)I_{k=0}^{n}f(k)+ (\nabla k=1)I_{k=0}^{n}g(k)$.

Proof: Suppose the primitive function of $f(n)$ is $F(n)$, that of $g(n)$ is $G(n)$, that of $f(n)+g(n)$ is $H(n)$. From Definition 20.3, we obtain

$H(0)=f(0)+g(0)-(f(0)+g(0))=(f(0)-f(0))+(g(0)-g(0))=F(0)+G(0)$,

$H(1)=f(0)+g(0)+f(1)+g(1)-(f(0)+g(0))=(f(0)+f(1)-f(0))+(g(0)+g(1)-g(0))=F(1)+G(1)$,

……

$H(n)=\sum_{k=0}^{n}(f(k)+g(k))-(f(0)+g(0))=\left(\sum_{k=0}^{n}f(k)-f(0)\right)+\left(\sum_{k=0}^{n}g(k)-g(0)\right)=F(n)+G(n)$.

Therefore, we have

$(\nabla k=1)I_{k=0}^{n}(f(k)+g(k))=H(n)=F(n)+G(n)= (\nabla k=1)I_{k=0}^{n}f(k)+ (\nabla k=1)I_{k=0}^{n}g(k)$.

Q.E.D.

Property 20.2: $(\nabla k=1)I_{k=0}^{n}(f(k)-g(k))= (\nabla k=1)I_{k=0}^{n}f(k)- (\nabla k=1)I_{k=0}^{n}g(k)$.

Proof: Similar to the proof of Property 20.1, so omitted here.

Property 20.3: $(\nabla k=1)I_{k=0}^{n}bf(k)=b(\nabla k=1)I_{k=0}^{n}f(k)$, where b is a constant.

Proof: Suppose the primitive function of $f(n)$ is $F(n)$, that of $bf(n)$ is $H(n)$. From Definition 20.3, we obtain

$H(0)=bf(0)-bf(0)=b(f(0)-f(0))=bF(0)$,

$H(1)=bf(0)+bf(1)-bf(0)=b(f(0)+f(1)-f(0))=bF(1)$,

……

$H(n)=\sum_{k=0}^{n}bf(k)-bf(0)=b\sum_{k=0}^{n}f(k)-bf(0)=b\sum_{k=0}^{n}f(k)-f(0)=bF(n)$.

Therefore, we have

$(\nabla k=1)I_{k=0}{}^{n}bf(k)=H(n)=bF(n)=b(\nabla k=1)I_{k=0}{}^{n}f(k)$.

Q.E.D.

20.1.2 Single-sided discrete definite integrals

20.1.2.1 Concept of single-sided discrete definite integrals

Example 20.3: Suppose we have a single-sided discrete function $m=f(n)=n$, shown in Fig. 20.1. Compute the sum of length of the vertical lines enclosed by $n=3$, $m=f(n)=n$, $n=6$, and n-axis.

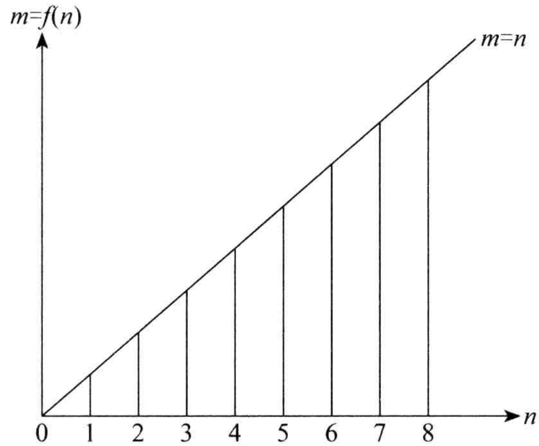


Fig. 20.1 Geometric interpretation of single-sided discrete definite integrals

Solution: The result is $\sum_{n=3}^{6}n=3+4+5+6=18$.

In this example, there are only 4 terms to sum, which can be done directly. If there are thousands of terms to sum, direct sum is infeasible. We must use primitive functions. For this reason, we transform (20.9) into the formula of single-sided discrete definite integrals. In Newtonian calculus, the integral of variable upper limit $\int_0^x f(t)dt$ can be written as $\int_0^x f(x)dx$. Likewise, $(\nabla k=1)I_{k=0}{}^{n}k$ of (20.9) can be written as $(\nabla n=1)I_{n=0}{}^{n}n$. And (20.9) becomes

$$(\nabla n=1)I_{n=0}{}^{n}n=n(n+1)/2|_0^n=n(n+1)/2-0(0+1)/2=n(n+1)/2. \quad (20.18)$$

Notice that now, the lower limit is b (b may not equals 0), the upper limit is no longer n, but c (b and c are natural numbers), formula (20.9) becomes

$$(\nabla n=1)I_{n=b}{}^{c}n=n(n+1)/2|_b^c \quad (20.19)$$

(every formula of single-sided discrete definite integrals of variable upper limit can be transformed into one of single-sided discrete definite integrals like this). As to this example, b=2, c=6, and we have

$$(\nabla n=1)I_{n=2}{}^{6}n=n(n+1)/2|_2^6=6(6+1)/2-2(2+1)/2=21-3=18.$$

Through Example 20.3, we learn the following: (1) The geometric interpretation of single-sided discrete definite integrals is the sum of length of the vertical lines enclosed by n=b+1, $f(n)$, n=c, and n-axis. (2) $(\nabla n=1)I_{n=b}{}^{c}f(n)=\sum_{n=b+1}^{c}f(n)$. (3) A single-sided discrete definite integral is computed by the upper limit of the primitive function minus the lower limit. (4) The transformation from formulas of single-sided discrete definite integrals of variable upper limit to formulas of single-sided discrete definite integrals.

In Newtonian calculus, the geometric interpretation of definite integrals is curved trapezoid; i.e., to sum, then to take limit of the differential of area. While the geometric interpretation of single-sided discrete definite integrals is not curved trapezoid, no need to introduce the concept of differential, and single-sided discrete definite integrals are not denoted as $I_{n=b}{}^{c}f(n)\nabla n$, but $(\nabla n=1)I_{n=b}{}^{c}f(n)$.

Definition 20.4: $\sum_{n=b+1}^{c}f(n)$ is called the single-sided discrete definite integral of $f(n)$ from b to c, denoted by $(\nabla n=1)I_{n=b}{}^{c}f(n)$.

From Definition 20.4 we learn

$$(\nabla n=1)I_{n=b}{}^{c}f(n)=\sum_{n=b+1}^{c}f(n). \tag{20.20}$$

Theorem 20.1: Suppose $F(n)$ is the primitive function of $f(n)$ and b, c$\in$**N**, c$>$b, then

$$(\nabla n=1)I_{n=b}{}^{c}f(n)=F(n)|_{b}{}^{c}=F(c)-F(b). \tag{20.21}$$

Proof: From Definition 20.3, we know

$F(b)=f(0)+f(1)+\ldots+f(b-1)+f(b)-f(0)$,

$F(c)=f(0)+f(1)+\ldots+f(b-1)+f(b)+f(b+1)+\ldots+f(c)-f(0)$.

Therefore,

$F(c)-F(b)=f(b+1)+\ldots+f(c)=\sum_{n=b+1}^{c}f(n)=(\nabla n=1)I_{n=b}{}^{c}f(n)$.

Q.E.D.

20.1.2.2 Properties of single-sided discrete definite integrals

Property 20.4: $(\nabla n=1)I_{n=b}{}^{b}f(n)=0$.

Proof: $(\nabla n=1)I_{n=b}{}^{b}f(n)=F(n)|_{b}{}^{b}=F(b)-F(b)=0$.

Q.E.D.

Property 20.5: Suppose b$<$d$<$c, then $(\nabla n=1)I_{n=b}{}^{c}f(n)=(\nabla n=1)I_{n=b}{}^{d}f(n)+(\nabla n=1)I_{n=d}{}^{c}f(n)$

Proof: $(\nabla n=1)I_{n=b}{}^{c}f(n)=\sum_{n=b+1}^{c}f(n)=\sum_{n=b+1}^{d}f(n)+\sum_{n=b+1}^{c}f(n)=(\nabla n=1)I_{n=b}{}^{d}f(n)+(\nabla n=1)I_{n=d}{}^{c}f(n)$.

Q.E.D.

20.1.3 Single-sided discrete left derivatives

20.1.3.1 Concept of single-sided discrete left derivatives

Example 20.4: Suppose we have a single-sided discrete function m=f(n)=n^2 shown in Fig. 20.2. What is the slope of the left tangent line of $m=n^2$ at n=3.

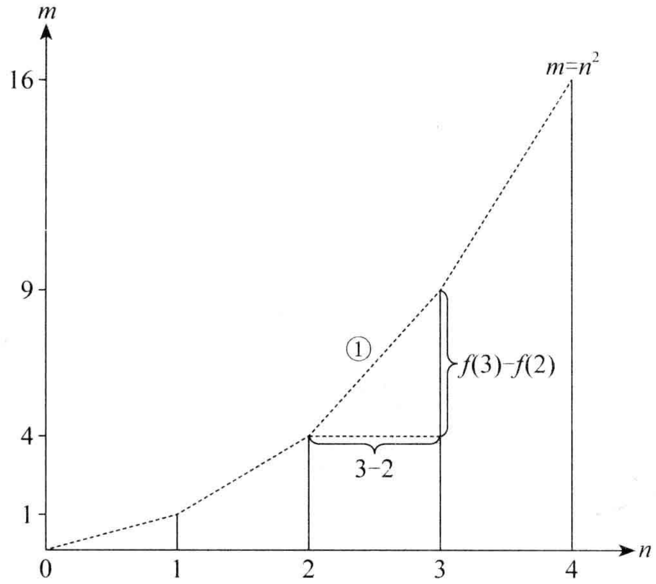


Fig. 20.2 Geometric interpretation of single-sided discrete left derivatives

Solution: The left tangent line of $m=n^2$ at $n=3$ is the line segment connecting (2, f(2)) and (3, f(3)), shown as the dotted line marked ① in Fig. 20.2. (the line segment has no definition except for the two ends, therefore, it is drawn with dotted line). The slope of the left tangent line is

$\{f(3)-f(2)\}/\{3-2\}=\{f(3)-f(2)\}/1=f(3)-f(2)=9-4=5.$

The geometric interpretation of the single-sided discrete left derivative of $f(n)$ at n_0 is the slope of the tangent line of $f(n)$ at n_0.

Definition 20.5: The single-sided discrete left derivative of $m=f(n)$ at n_0 is the backward difference quotient of $m=f(n)$ at n_0:

$$\frac{\nabla m}{\nabla n}\Big|_{n=n0}=\frac{\nabla f}{\nabla n}\Big|_{n=n0}=\{f(n_0)-f(n_0-1)\}/\{n_0-(n_0-1)\}=\{f(n_0)-f(n_0-1)\}/1=f(n_0)-f(n_0-1)=\nabla f(n_0),$$

where $\nabla f(n_0)$ is the backward difference of $f(n)$ at n_0.

Definition 20.6: Suppose the single-sided discrete left derivative of $m=f(n)$ exist except for $n=0$, then we say that the left derivative function of $m=f(n)$ exist, denoted by

$$\frac{\nabla m}{\nabla n}=\frac{\nabla f}{\nabla n}=\{f(n)-f(n-1)\}/\{n-(n-1)\}=\{f(n)-f(n-1)\}/1=f(n)-f(n-1)=\nabla f(n).$$

Example 20.5: What is the left derivative function of $m=n^2$?

Solution: There is no left derivative of $m=n^2$ at $n=0$, the left derivative at $n=1$ is $1^2-0^2=1$, at $n=2$ is $2^2-1^2=3$, at $n=3$ is $3^2-2^2=5$, at $n=4$ is $4^2-3^2=7$, ... We can induce that the left derivative function of $m=n^2$ is

$\nabla n^2/\nabla n=2n-1.$

If no confusion occurs, we simply call the left derivative function left derivative.

20.1.3.2 Relationship between left derivatives and primitive functions

Theorem 20.2: Suppose $(\nabla k=1)I_{k=0}^{n}f(k)=F(n)$, then $\frac{\nabla F(n)}{\nabla n}=f(n)$.

Proof: From $(\nabla k=1)I_{k=0}^{n}f(k)=F(n)$ and Definition 20.3 we know

$F(0)=f(0)-f(0)$,

$F(1)=\sum_{k=0}^{1}f(k)-f(0)$,

$F(2)=\sum_{k=0}^{2}f(k)-f(0)$,

……

$F(n)=\sum_{k=0}^{0}f(k)-f(0)$.

Now, we compute the left derivative of $F(n)$, obtaining

$$\frac{\nabla F}{\nabla n}\bigg|_{n=1}=\nabla F(1)=F(1)-F(0)=f(1),$$

$$\frac{\nabla F}{\nabla n}\bigg|_{n=2}=\nabla F(2)=F(2)-F(1)=f(2),$$

……

$$\frac{\nabla F}{\nabla n}\bigg|_{n=n}=\nabla F(n)=F(n)-F(n-1)=f(n).$$

That is,

$$\frac{\nabla F(n)}{\nabla n}=f(n).$$

Q.E.D.

From $(\nabla k=1)I_{k=0}^{n}f(k)=F(n)$ and $\frac{\nabla F(n)}{\nabla n}=f(n)$ we know the relationship between the left derivative and the primitive function: the single-sided discrete integral of $f(k)$ is $F(n)$, the single-sided discrete left derivative of $F(n)$ is $f(n)$.

20.1.3.3 Properties of single-sided discrete left derivatives

Property 20.6: $\frac{\nabla}{\nabla n}(f_1(n)+f_2(n))=\nabla f_1(n)/\nabla n+\nabla f_2(n)/\nabla n$.

Proof: Do not lose generality, suppose n is any natural number on the n-axis except 0.

$$\frac{\nabla}{\nabla n}(f_1(n)+f_2(n))$$
$$=\nabla(f_1(n)+f_2(n))$$
$$=f_1(n)+f_2(n)-(f_1(n-1)+f_2(n-1))$$
$$=(f_1(n)-f_1(n-1))+(f_2(n)-f_2(n-1))$$
$$=\nabla f_1(n)+\nabla f_2(n)$$
$$=\nabla f_1(n)/\nabla n+\nabla f_2(n)/\nabla n.$$

Q.E.D.

Property 20.7: $\frac{\nabla}{\nabla n}(f_1(n)-f_2(n))=\nabla f_1(n)/\nabla n-\nabla f_2(n)/\nabla n$.

Proof: Similar to the proof of Property 20.6, so omitted.

Property 20.8: $\frac{\nabla bf(n)}{\nabla n}=b\frac{\nabla f(n)}{\nabla n}$, where b is a constant.

Proof: $\frac{\nabla bf(n)}{\nabla n}=\nabla(bf(n))=bf(n)-bf(n-1)=b(f(n)-f(n-1))=b\nabla f(n)=b\frac{\nabla f(n)}{\nabla n}$.

Q.E.D.

20.1.3.4 Deduction of formulas of single-sided discrete left derivatives

The following are some of the important formulas of single-sided discrete left derivatives.

Single-sided discrete left derivatives of exponential functions:

$$\nabla(a^n)/\nabla n=(a-1)a^{n-1} \tag{20.22}$$

Single-sided discrete left derivatives of power functions:

$$\frac{\nabla b}{\nabla n}=0 \text{ (b is a constant)} \tag{20.23}$$

$$\frac{\nabla n}{\nabla n}=1 \tag{20.24}$$

$$\nabla(n^2)/\nabla n=2n-1 \tag{20.25}$$

$$\nabla(n^3)/\nabla n=3n^2-3n+1 \tag{20.26}$$

$$\nabla(n^4)/\nabla n=4n^3-6n^2+4n-1 \tag{20.27}$$

$$\nabla(n^5)/\nabla n=5n^4-10n^3+10n^2-5n+1 \tag{20.28}$$

……

$$\nabla(n^b)/\nabla n=(-1)^0\binom{b}{1}n^{b-1}+(-1)^1\binom{b}{2}n^{b-2}+\ldots+(-1)^{b-1}\binom{b}{b}n^0, \text{ where } b\in\mathbf{N} \tag{20.29}$$

Single-sided discrete left derivatives of products:

$$\nabla(na^n)/\nabla n=\frac{\nabla n}{\nabla n}a^n+n(\nabla(a^n)/\nabla n)-(a-1)a^{n-1} \tag{20.30}$$

$$\nabla(n^2a^n)/\nabla n=(\nabla(n^2)/\nabla n)a^n+n^2(\nabla(a^n)/\nabla n)-(a-1)na^n+(a-1)[(a-2)n+1]a^{n-1} \tag{20.31}$$

Prove (20.23):

Do not lose generality, suppose *n* is any point on the *n*-axis except 0, we have

$$\left.\frac{\nabla b}{\nabla n}\right|_{n=n}=\frac{b-b}{n-(n-1)}=0.$$

Q.E.D.

Prove (20.22):

We change (20.7) into

$$(\nabla k=1)I_{k=0}^{n-1}a^k=\{a^n-1\}/\{a-1\}-1. \tag{20.32}$$

Both sides of (20.32) times (a−1), obtain

$$(\nabla k=1)I_{k=0}^{n-1}(a-1)a^k=a^n-a. \tag{20.33}$$

We take single-sided discrete left derivative on n to both sides of (20.33), according to Theorem 20.2, obtaining

$(a-1)a^{n-1}=\nabla(a^n)/\nabla n.$

Q.E.D.

Prove (20.24):

We rewrite (20.8) as follows:

$(\nabla k=1)I_{k=0}{}^{n}1=n.$

We take single-sided discrete left derivative on n to both sides of (20.8), according to Theorem 20.2, obtaining

$$1=\frac{\nabla n}{\nabla n}.$$

Q.E.D.

Prove (20.25):

We rewrite (20.9) as follows:

$(\nabla k=1)I_{k=0}{}^{n}k=\{n^2+n\}/2.$

Both sides of (20.9) times 2, obtain

$$(\nabla k=1)I_{k=0}{}^{n}2k=n^2+n. \tag{20.34}$$

We take single-sided discrete left derivative on n to both sides of (20.34), according to Theorem 20.2, obtaining

$$2n=\nabla(n^2)/\nabla n+\nabla n/\nabla n=\nabla(n^2)/\nabla n+1. \tag{20.35}$$

Shift terms, obtaining

$\nabla(n^2)/\nabla n=2n-1.$

Q.E.D.

The proofs of (20.26) to (20.28) are similar to that of (20.25). Observe (20.24) to (20.28), we can find that their coefficients form a Pascal triangle of alternate positive and negative sign. Therefore, it is easy for us to induce (20.29).

Prove (20.30):

We change (20.13) into

$(\nabla k=1)I_{k=0}{}^{n-1}ka^k=\{[(a-1)(n-1)-1]a^n+a\}/(a-1)^2.$

Both sides of (20.13) times $(a-1)^2$, obtain

$$(\nabla k=1)I_{k=0}{}^{n-1}(a-1)^2ka^k=[(a-1)(n-1)-1]a^n+a=(a-1)na^n-aa^n+a. \tag{20.36}$$

We take single-sided discrete left derivative on n to both sides of (20.36), according to Theorem 20.2, obtaining

$$(a-1)^2(n-1)a^{n-1}=(a-1)(\nabla(na^n)/\nabla n)-a(\nabla(a^n)/\nabla n). \tag{20.37}$$

Both sides of (20.37) are divided by $(a-1)$, and

$(a-1)(n-1)a^{n-1}$

$$=\nabla(na^n)/\nabla n-\frac{a-1+1}{a-1}(\nabla(a^n)/\nabla n)$$

$$=\nabla(na^n)/\nabla n-\nabla(a^n)/\nabla n-\frac{1}{a-1}(\nabla(a^n)/\nabla n)$$

$$=\nabla(na^n)/\nabla n-\nabla(a^n)/\nabla n-a^{n-1} \tag{20.38}$$

is obtained. Shift terms for (20.38), obtaining

$$\nabla(na^n)/\nabla n$$

$$=(a-1)(n-1)a^{n-1}+\nabla(a^n)/\nabla n+a^{n-1}$$

$$=(n-1)(\nabla(a^n)/\nabla n)+\nabla(a^n)/\nabla n+a^{n-1}$$

$$=n(\nabla(a^n)/\nabla n)+a^{n-1}. \tag{20.39}$$

Formula (20.30) is

$$\nabla(na^n)/\nabla n$$

$$=\frac{\nabla n}{\nabla n}a^n+n(\nabla(a^n/\nabla n)-(a-1)a^{n-1}$$

$$=a^n+n(\nabla(a^n)/\nabla n)-a^n+a^{n-1}$$

$$=n(\nabla(a^n)/\nabla n)+a^{n-1} \tag{20.40}$$

Comparing the right sides of (20.39) and (20.40), we see that they equal each other, therefore (20.30) is proved.

The proof of (20.31) is similar to that of (20.30).

20.1.3.5 Higher-order single-sided discrete left derivatives

Definition 20.7: Second-order single-sided discrete left derivative is defined as the second-order backward difference quotient:

$$\nabla^2 f/\nabla n^2$$

$$=\left\{\frac{\nabla f(n)}{\nabla n}-\frac{\nabla f(n-1)}{\nabla n}\right\}/\{n-(n-1)\}=\frac{f(n)-f(n-1)-(f(n-1)-f(n-2)}{\nabla n}$$

$$=f(n)-2f(n-1)+f(n-2)$$

$$=\nabla^2 f(n).$$

Second-order backward difference quotient is the same in value as second-order backward difference.

Example 20.6: Suppose $m=n^2$, compute $\nabla^2 m/\nabla n^2$.

Solution: $\frac{\nabla m}{\nabla n}=2n-1$, $\nabla^2 m/\nabla n^2=\frac{\nabla}{\nabla n}\left(\frac{\nabla m}{\nabla n}\right)=2.$

20.1.4 Single-sided discrete indefinite integrals

Suppose $f(n)$ has a primitive function $F(n)$; i.e., $\frac{\nabla F}{\nabla n}=f(n)$, then for any constant C, we have

$$\frac{\nabla(F(n)+C)}{\nabla n}=\frac{\nabla F(n)}{\nabla n}+\frac{\nabla C}{\nabla n}=\frac{\nabla F(n)}{\nabla n}+0=\frac{\nabla F(n)}{\nabla n}=f(n).$$

That is, for any constant C, $F(n)$+C is also a primitive function of $f(n)$. This shows, if $f(n)$ has one primitive function, then it has infinitely many primitive functions.

On the other hand, if $G(n)$ is another primitive function of $f(n)$; i.e., $\frac{\nabla G}{\nabla n}=f(n)$. Then

$$\frac{\nabla(G(n)-F(n))}{\nabla n}=\frac{\nabla G(n)}{\nabla n}-\frac{\nabla F(n)}{\nabla n}=f(n)-f(n)=0.$$

According to (20.23), we have

$G(n)-F(n)=C_0$, (C_0 is a constant).

This shows, the difference of $G(n)$ and $F(n)$ is a constant. Therefore, when C is an arbitrary constant,

$F(n)+C$

can denote any primitive function of $f(n)$.

Definition 20.8: Suppose $F(n)$+C is any primitive function of $f(n)$, then

$(\nabla n=1)\mathrm{I}f(n)=F(n)+C$

is called the single-sided discrete indefinite integral of $f(n)$, where I is the integral sign, $f(n)$ is the integrand, n is the integral variable, $F(n)$+C is the primitive function.

Formulas of single-sided discrete indefinite integral can be obtained from those of single-sided discrete definite integral of variable upper limit by eliminating the lower and upper limits from the left side and adding constant C to the right side. For example, (20.9) can be transformed into

$$(\nabla n=1)\mathrm{I}n=\{n^2+n\}/2+C \tag{20.41}$$

20.2 Single-sided discrete calculus of binary functions

Definition 20.9: The single-sided discrete left partial derivative of $l=f(m, n)$ to m at (m_0, n_0) is defined as

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=m0,\ n=n0}=\{f(m_0, n_0)-f(m_{-1}, n_0)\}/\{m_0-m_{-1}\}.$$

$\frac{\nabla l}{\nabla m}(m, n)|_{m=m0,\ n=n0}$ can be defined similarly.

Example 20.7: Suppose $l=mn^2$, compute $\frac{\nabla l}{\nabla m}(m, n)|_{m=3,\ n=4}$.

Solution: We use the matrix shown in Fig. 20.3 to denote $l=mn^2$.

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} n\,0 & 1 & 2 & 3 & 4 \end{array} \\
m & \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 9 & 16 \\ 0 & 2 & 8 & 18 & 32 \\ 0 & 3 & 12 & 27 & 48 \\ 0 & 4 & 16 & 36 & 64 \end{bmatrix}
\end{array}
$$

Fig. 20.3 Matrix representation of $l=mn^2$

In Fig. 20.3, the column index is *m*, the row index is *n*, the matrix element is *l*. For example, when *m*=3 and *n*=4, then *l*=48. From Fig. 20.3, we learn

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=3,\, n=4}=\{f(m_3, n_4)-f(m_2, n_4)\}/\{m_3-m_2\}=\frac{48-32}{3-2}=16.$$

Definition 20.10: If for any (*m*, *n*), $\frac{\nabla l}{\nabla m}(m, n)$ exists, then $\frac{\nabla l}{\nabla m}(m, n)$ is called the single-sided discrete left partial derivative function of *f* to *m*.

$\frac{\nabla l}{\nabla m}(m, n)$ can be defined similarly. When no confusion occurs, single-sided discrete left partial derivative function is simply called single-sided discrete left partial derivative.

When computing $\frac{\nabla l}{\nabla m}(m, n)$, *n* is kept constant. When computing $\frac{\nabla l}{\nabla m}(m, n)$, *m* is kept constant.

Example 20.8: Suppose $l=mn^2$, compute $\frac{\nabla l}{\nabla m}(m, n)$ and $\frac{\nabla l}{\nabla n}(m, n)$.

Solution: $\frac{\nabla l}{\nabla m}(m, n)=n^2$, $\frac{\nabla l}{\nabla n}(m, n)=m(2n-1)$.

20.2.2 Single-sided discrete double integrals

Definition 20.11: Suppose $l=f(m, n)$, then

$$(\nabla m=1)I_{m=c}^{d}(\nabla n=1)I_{n=a}^{b}f(m, n)$$

is called the single-sided discrete double integral, and when integrating on *n*, *m* is kept constant.

Example 20.9: Compute $(\nabla m=1)I_{m=3}^{7}(\nabla n=1)I_{n=2}^{5}m$.

Solution:

$$
\begin{aligned}
&(\nabla m=1)I_{m=3}^{7}(\nabla n=1)I_{n=2}^{5}m \\
&=(\nabla m=1)I_{m=3}^{7}(mn)|_{2}^{5} \\
&=(\nabla m=1)I_{m=3}^{7}m(5-2) \\
&=(\nabla m=1)I_{m=3}^{7}3m \\
&=3\{m^2+m\}/2|_{3}^{7} \\
&=\frac{3}{2}(7^2+7-3^2-3) \\
&=66.
\end{aligned}
$$

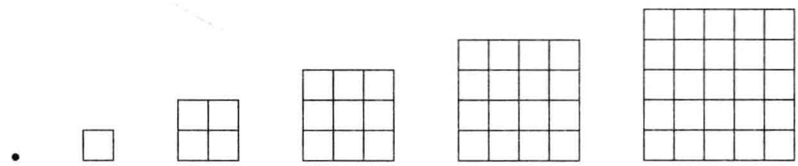
20.3 Single-sided discrete ordinary differential equations

An equation containing an unknown single-sided discrete unary function and some of its left derivatives is called a single-sided discrete ordinary differential equation. Solving the equation is to find the unary function. Given the equation and its initial conditions to find the particular solution is called the initial value problem of the equation.

20.3.1 Simple single-sided discrete ordinary differential equations

A simple single-sided discrete ordinary differential equation is one that only the highest-order left derivative exists.

Example 20.10: Given the side n of a square, the area n^2 of the square is shown in Fig. 20.4.



n	0	1	2	3	4	5
$m=n^2$	0	1	4	9	16	25
$\dfrac{\nabla m}{\nabla n}$		1	3	5	7	9
$\nabla^2 m/\nabla n^2$			2	2	2	2

Fig. 20.4 Example of single-sided discrete ordinary differential equation

The first row in Fig. 20.4 is the sides of the square, the second row is the areas, the third row is the first-order single-sided discrete left derivatives of area to side, the fourth row is the second-order left derivatives. The initial value problem is

$$\nabla^2 m/\nabla n^2=2 \tag{20.42}$$

$$m|_{n=0}=0 \tag{20.43}$$

$$\frac{\nabla m}{\nabla n}\Big|_{n=1}=1 \tag{20.44}$$

In this example, the function is known: $m=n^2$. Our aim is to compute the function from the initial value problem, to verify the correctness of single-sided discrete ordinary differential equation theory. Later examples in this chapter are the same.

Solution: We take single-sided discrete integral on n to both sides of (20.42), obtaining

$$(\nabla n=1)\mathrm{I}(\nabla^2 m/\nabla n^2)=(\nabla n=1)\mathrm{I}2$$

and

$$\frac{\nabla m}{\nabla n}=2n+C_1 \tag{20.45}$$

Substitute (20.44) into (20.45), obtaining

$$1=2*1+C_1\,,\ C_1=-1$$

and

$$\frac{\nabla m}{\nabla n}=2n-1 \qquad (20.46)$$

We take single-sided discrete integral on *n* to both sides of (20.46), obtaining

$$(\nabla n=1)I\frac{\nabla m}{\nabla n}=(\nabla n=1)I(2n-1)$$

and

$$m=2\{n^2+n\}/2-n+C_2=n^2+C_2 \qquad (20.47)$$

Substitute (20.43) into (20.47), obtaining

$$0=0^2+C_2,\ C_2=0$$

and

$$m=n^2. \qquad (20.48)$$

20.3.2 Single-sided discrete second-order linear constant-coefficients homogeneous ordinary differential equations

A single-sided discrete second-order linear constant-coefficients homogeneous ordinary differential equation is in the form of:

$$\nabla^2 m/\nabla n^2+p\frac{\nabla m}{\nabla n}+qm=0 \qquad (20.49)$$

Suppose the solution is $m=a^n$, then we have

$$\frac{\nabla m}{\nabla n}=(a-1)a^{n-1} \qquad (20.50)$$

and

$$\nabla^2 m/\nabla n^2=(a-1)(a-2)a^{n-2} \qquad (20.51)$$

Substitute (20.50) and (20.51) into (20.49), obtaining

$$\begin{aligned}&(a-1)(a-2)a^{n-2}+p(a-1)a^{n-1}+qa^n\\&=(a-1)(a-2)a^{n-2}+pa(a-1)a^{n-2}+qa^2a^{n-2}\\&=a^{n-2}[(a-1)(a-2)+pa(a-1)+qa^2]\\&=0\end{aligned} \qquad (20.52)$$

Because a^{n-2} is not equal to 0, formula (20.52) becomes

$$(a-1)(a-2)+pa(a-1)+qa^2=0 \qquad (20.53)$$

and

$$(p+q+1)a^2-(p+3)a+2=0 \qquad (20.54)$$

Solve (20.54) for a, obtaining

$$a_{1,2}=\frac{p+3\pm\sqrt{[-(p+3)]^2-4*2*(p+q+1)}}{2(p+q+1)}$$

Suppose a_1 and a_2 are two distinct real numbers, then $m_1=a_1{}^n$ and $m_2=a_2{}^n$ are two particular solutions of (20.49) and $m=C_1a_1{}^n+C_2a_2{}^n$ is a general solution of (20.49).

20.4 Single-sided discrete partial differential equations

Example 20.11: Suppose *l=mn* shown in Fig. 20.5.

$$\begin{array}{c|c} & n\quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \\ m & \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 6 & 8 \\ 0 & 3 & 6 & 9 & 12 \\ 0 & 4 & 8 & 12 & 16 \end{bmatrix} \end{array}$$

Fig. 20.5 Matrix representation of *l=mn*

We have $\frac{\nabla l}{\nabla m}(m, n)=n$, $\nabla^2 l/\nabla m\nabla n(m, n)=1$. The initial value problem is:

$$\nabla^2 l/\nabla m\nabla n(m, n)=1 \tag{20.55}$$

$$l|_{m=0,\ n=0}=0 \tag{20.56}$$

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=1,\ n=1}=1 \tag{20.57}$$

Solution: Integrate on *n* to both sides of (20.55), obtaining

$$(\nabla n=1)I\nabla^2 l/\nabla m\nabla n(m, n)=(\nabla n=1)I1$$

and

$$\frac{\nabla l}{\nabla m}(m, n)=n+C_1 \tag{20.58}$$

Substitute (20.57) into (20.58), obtaining

$$C_1=0$$

and

$$\frac{\nabla l}{\nabla m}(m, n)=n \tag{20.59}$$

Integrate on *m* to both sides of (20.59), obtaining

$$(\nabla m=1)I\frac{\nabla l}{\nabla m}(m, n)=(\nabla m=1)In$$

and

$$l=mn+C_2 \tag{20.60}$$

Substitute (20.56) into (20.60), obtaining

$$C_2=0$$

and

l=mn.

Chapter 21 Unified calculus

21.1 Overview of this chapter

Unified calculus unifies single-sided discrete calculus and Newtonian calculus. In single-sided discrete calculus, the arguments range over natural numbers (i.e., the distance between two points in the axes of the arguments h=1), the function value ranges over real numbers. In Newtonian calculus, the arguments and the function value all range over real numbers ($h \to 0$). In unified calculus, the distance between two points in the axes of the arguments h can be any number with h=1 and $h \to 0$ being two special cases, the function value ranges over real numbers.

Unified calculus is different from time scales calculus constructed by Hilger in that the former unifies single-sided discrete calculus including single-sided discrete differential equation and Newtonian calculus including continuous differential equation, the latter unifies difference equation and continuous differential equation. The similarities of the two calculi are shown in Table 21.1.

Table 21.1 Similarities between time scales calculus and unified calculus

Time scales calculus	Unified calculus
T=R	$h \to 0$
T=Z	h=1
T=hZ	h is any number

The differences between time scales calculus and unified calculus are:

(1) Time scales calculus adopts forward difference, while unified calculus adopts backward difference quotient.

(2) When T=hZ, the integral of time scales calculus is defined as

$$\int_a^b f(t)\Delta t=\sum_{k=a/h}^{b/h-1} f(kh)h \qquad \text{if } a<b,$$

which means that the geometric interpretation of the integral of time scales calculus is the area of the curved trapezoid enclosed by $k=\frac{a}{h}$, f(kh), $k=\frac{b}{h}-1$, and t-axis, and $f(kh)h$ is a rectangle of the curved trapezoid, $f(kh)$ is the height of the rectangle, h is the width of it. The integral of unified calculus is defined as

$$(\nabla n=h)\mathrm{I}_{n=a}{}^{b}f(n),$$

which means that the geometric interpretation of the integral of unified calculus is the sum of the length of the vertical lines enclosed by n=a+h, $f(n)$, n=b, and n-axis, and the distance between two vertical lines is h.

(3) In time scales calculus, when T=Z, the domain is **Z**. In unified calculus, when h=1, the domain is **N**.

In Chapters 19 and 20, the formulas of derivatives and integrals are obtained by deductive inference. In this chapter, the formulas of integrals are obtained by inductive inference; the formulas of derivatives are obtained from

$$\frac{f(n)-f(n-\nabla n)}{\nabla n}\Big|_{\nabla n=h}=\frac{f(n)-f(n-h)}{h}.$$

When h=1, it becomes the single-sided discrete left derivative, when $h\to 0$, it becomes Newtonian derivative.

21.2 Unified calculus of unary functions

21.2.1 Power functions

21.2.1.1 Unified left derivatives of power functions

$$\frac{\nabla n}{\nabla n}\Big|_{\nabla n=h}=1 \tag{21.1}$$

When h=1, formula (21.1) becomes $\frac{\nabla n}{\nabla n}=1$, the same as (20.24). When $h\to 0$, formula (21.1) becomes $\frac{dx}{dx}=1$.

$$\nabla(n^2)/\nabla n|_{\nabla n=h}=2n-h \tag{21.2}$$

When h=1, formula (21.2) becomes $\nabla(n^2)/\nabla n=2n-1$, the same as (20.25). When $h\to 0$, formula (21.2) becomes $d(x^2)/dx=2x$.

$$\nabla(n^3)/\nabla n|_{\nabla n=h}=3n^2-3nh+h^2 \tag{21.3}$$

When h=1, formula (21.3) becomes $\nabla(n^3)/\nabla n=3n^2-3n+1$, the same as (20.26). When $h\to 0$, formula (21.3) becomes $d(x^3)/dx=3x^2$.

$$\nabla(n^4)/\nabla n|_{\nabla n=h}=4n^3-6n^2h+4nh^2-h^3 \tag{21.4}$$

When h=1, formula (21.4) becomes $\nabla(n^4)/\nabla n=4n^3-6n^2+4n-1$, the same as (20.27). When $h\to 0$, formula (21.4) becomes $d(x^4)/dx=4x^3$.

$$\nabla(n^5)/\nabla n|_{\nabla n=h}=5n^4-10n^3h+10n_2h^2-5nh^3+h^4 \tag{21.5}$$

When h=1, formula (21.5) becomes $\nabla(n^5)/\nabla n=5n^4-10n^3+10n^2-5n+1$, the same as (20.28). When $h\to 0$, formula (21.5) becomes $d(x^5)/dx=5x^4$.

To sum up, the unified left derivatives of power functions treat both single-sided discrete left derivatives and Newtonian derivatives as its special cases.

21.2.1.2 Unified integrals of power functions

$$(\nabla k=h)I_{k=0}^{n}1=\frac{n}{h} \tag{21.6}$$

When h=1, formula (21.6) becomes $(\nabla k=1)I_{k=0}^{n}1=n$, the same as (20.8). When $h\to 0$, we continuize (continuize is a new English word coined by the author, meaning the opposite of discretize) formula (21.6): When the distance h between two adjacent points on the argument axis approaches 0, the discrete variable n becomes the continuous variable x; i.e., $\lim_{h\to 0}\frac{n}{h}=x$, obtaining $\int_0^x 1dt=x$.

$$(\nabla k=h)I_{k=0}^{n}k=\left\{\left(\frac{n}{h}\right)^2+\frac{n}{h}\right\}/\left(\frac{2}{h}\right) \tag{21.7}$$

When h=1, formula (21.7) becomes $(\nabla k=1)I_{k=0}^{n}k=\{n^2+n\}/2$, the same as (20.9). When $h\to 0$, we continuize (21.7): We retain only the highest order of infinity, obtaining

$$(\nabla k=h\to 0)I_{k=0}^{n}k=\left(\frac{n}{h}\right)^2/\left(\frac{2}{h}\right) \tag{21.8}$$

In (21.8), when $h\to 0$, k becomes t, n becomes x, 2 is kept constant; i.e.,

$$\lim_{h\to 0}\left(\frac{n}{h}\right)^2/\left(\frac{2}{h}\right)=x^2/2,$$

obtaining $\int_0^x tdt=x=x^2/2$.

$$(\nabla k=h)I_{k=0}^{n}k^2=\left\{2\left(\frac{n}{h}\right)^3+3\left(\frac{n}{h}\right)^2+\right\}/(6/h^2) \tag{21.9}$$

When h=1, formula (21.9) becomes $(\nabla k=1)I_{k=0}^{n}k^2=\{2n^3+3n^2+n\}/6=n(n+1)(n+2)/6$, the same as (20.10). When $h\to 0$, we continuize (21.9), obtaining $\int_0^x t^2dt=x^3/3$.

$$(\nabla k=h)I_{k=0}^{n}k^3=\left\{\left(\frac{n}{h}\right)^4+2\left(\frac{n}{h}\right)^3+\left(\frac{n}{h}\right)^2\right\}/(4/h^3) \tag{21.10}$$

When h=1, formula (21.10) becomes $(\nabla k=1)I_{k=0}^{n}k^3=\{n^4+2n^3+n^2\}/4=n^2(n+1)^2/4$, the same as (20.11). When $h\to 0$, we continuize (21.10), obtaining $\int_0^x t^3dt=x^4/4$.

$$(\nabla k=h)I_{k=0}^{n}k^4=\left\{6\left(\frac{n}{h}\right)^5+15\left(\frac{n}{h}\right)^4+10\left(\frac{n}{h}\right)^3-\left(\frac{n}{h}\right)\right\}/(30/h^4) \tag{21.11}$$

When h=1, formula (21.11) becomes $(\nabla k=1)I_{k=0}^{n}k^4=\{6n^5+15n^4+10n^3-n\}/30$, the same as (20.12). When $h\to 0$, we continuize (21.11), obtaining $\int_0^x t^4dt=x^5/5$.

To sum up, after introducing continuization, the unified integral of power functions treats both single-sided discrete integrals and Newtonian integrals as its special cases.

21.2.2 Unified left derivatives of exponential functions

$$\nabla(a^n)/\nabla n|_{\nabla n=h}=\{a^h-1\}\div h*(a^{n-h}) \tag{21.12}$$

When h=1, formula (21.12) becomes $\nabla(a^n)/\nabla n=(a-1)a^{n-1}$, the same as (20.22). When $h\to 0$, notice that $\lim_{h\to 0}\{a^h-1\}/h=\ln a$, formula (21.12) becomes $d(a^x)/dx=(\ln a)a^x$.

To sum up, the unified left derivative of exponential function unifies both single-sided discrete left derivative and Newtonian derivative.

21.2.3 Unified left derivatives of products of functions

$$\nabla(na^n)/\nabla n|_{\nabla n=h}=\frac{\nabla n}{\nabla n}a^n+n(\nabla(a^n)/\nabla n)-(a^h-1)a^{n-h} \tag{21.13}$$

When $h=1$, formula (21.13) becomes $\nabla(na^n)/\nabla n=\frac{\nabla n}{\nabla n}a^n+n(\nabla(a^n)/\nabla n)-(a-1)a^{n-1}$, the same as (20.30). When $h\to 0$, formula (21.13) becomes $d(xa^x)/dx=\frac{dx}{dx}a^x+x(d(a^x)/dx)=a^x+x*\ln a*a^x$.

$$\nabla(n^2a^n)/\nabla n|_{\nabla n=h}=(\nabla(n^2)/\nabla n)a^n+n^2(\nabla(a^n)/\nabla n)-(a^h-1)na^n+(a^h-1)[(a^h-2)n+h]a^{n-h} \tag{21.14}$$

When $h=1$, formula (21.14) becomes $\nabla(n^2a^n)/\nabla n|=(\nabla(n^2)/\nabla n)a^n+n^2(\nabla(a^n)/\nabla n)-(a-1)na^n+(a-1)[(a-2)n+1]a^{n-1}$, the same as (20.31). When $h\to 0$, formula (21.14) becomes $d(x^2a^x)/dx=(d(x^2)/dx)a^x+x^2(d(a^x)/dx=2xa^x+x^2*\ln a*a^x$.

To sum up, the unified left derivatives of products of functions unifies both single-sided discrete left derivatives and Newtonian derivatives.

21.2.4 Unified left derivatives of negative power functions

The single-sided discrete left derivative of this section and next section are obtained by inductive inference.

$$\nabla\left(\frac{1}{n}\right)/\nabla n|_{\nabla n=1}=-\frac{1}{(n-1)n} \tag{21.15}$$

$$\nabla\left(\frac{1}{n}\right)/\nabla n|_{\nabla n=h}=-\frac{1}{(n-h)n} \tag{21.16}$$

When $h=1$, formula (21.16) becomes (21.15). When $h\to 0$, formula (21.16) becomes $d\left(\frac{1}{x}\right)/dx=-1/x^2$.

$$\nabla(1/n^2)/\nabla n|_{\nabla n=1}=-\{2n-1\}/(n-1)^2n^2 \tag{21.17}$$

$$\nabla(1/n^2)/\nabla n|_{\nabla n=h}=-\{2n-h\}/(n-h)^2n^2 \tag{21.18}$$

When $h=1$, formula (21.18) becomes (21.17). When $h\to 0$, formula (21.18) becomes $d(1/x^2)/dx=-2/x^3$.

21.2.5 Unified left derivatives of quotients of functions

$$\nabla(a^n/n)/\nabla n|_{\nabla n=1}=\{(a-1)a^{n-1}n-a^n\}/(n-1)n \tag{21.19}$$

$$\nabla(a^n/n)/\nabla n|_{\nabla n=h}=\{(a^h-1)\div h*a^{n-h}*n-a^n\}/(n-h)n \tag{21.20}$$

When $h=1$, formula (21.20) becomes (21.19). When $h\to 0$, formula (21.20) becomes $d(a^x/x)/dx=\{(\ln a)a^x x-a^x\}/x^2$.

$$\nabla(a^n/n^2)/\nabla n|_{\nabla n=1}=\{(a-1)a^{n-1}n^2-a^n(2n-1)\}/(n-1)^2n^2 \tag{21.21}$$

$$\nabla(a^n/n^2)/\nabla n|_{\nabla n=h}=\{(a^h-1)\div h*a^{n-h}*n^2-a^n(2n-h)\}/(n-h)^2n^2 \tag{21.22}$$

When h=1, formula (21.22) becomes (21.21). When $h \to 0$, formula (21.22) becomes $d(a^x/x^2)/dx=\{(\ln a)a^x x^2-a^x 2x\}/x^4=\{(\ln a)a^x x-2a^x\}/x^3$.

To sum up, the unified left derivatives of quotients of functions treat both single-sided discrete left derivatives and Newtonian derivatives as its special cases.

21.3 Unified calculus of binary functions

21.3.1 Unified left partial derivatives

Definition 21.1: The unified left partial derivative of $l=f(m, n)|_{\nabla m=h1,\ \nabla n=h2}$ on m at (m_0, n_0) is defined as

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=m0,\ n=n0,\ \nabla m=h1,\ \nabla n=h2}=\{f(m_0, n_0)-f(m_0-h_1, n_0)\}/h_1.$$

Example 21.1: Suppose $l=mn^2|_{\nabla m=3,\ \nabla n=2}$, compute $\frac{\nabla l}{\nabla m}(m, n)|_{m=3,\ n=2,\ \nabla m=3,\ \nabla n=2}$.

Solution: $l=mn^2|_{\nabla m=3,\ \nabla n=2}$ is shown in Fig. 21.1.

$$\begin{array}{c|c}
 & n\ \ 0\ \ \ 2\ \ \ 4\ \ \ 6 \\
m & \\
\begin{array}{c}0\\3\\6\\9\end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 12 & 48 & 108 \\ 0 & 24 & 96 & 216 \\ 0 & 36 & 144 & 324 \end{bmatrix}
\end{array}$$

Fig. 21.1 Matrix representation of $l=mn^2|_{\nabla m=3,\ \nabla n=2}$

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=3,\ n=2,\ \nabla m=3,\ \nabla n=2}=\frac{12-0}{3}=4.$$

$\frac{\nabla l}{\nabla n}(m, n)|_{m=m0,\ n=n0,\ \nabla m=h1,\ \nabla n=h2}$ can be similarly defined.

21.3.2 Unified left partial derivative functions

Suppose $l=f(m, n)|_{\nabla m=h1,\ \nabla n=h2}$. When we compute the unified left derivative function $\frac{\nabla l}{\nabla n}(m, n)|_{\nabla m=h1,\ \nabla n=h2}$ of l on n, we keep m constant. When we compute the unified left derivative function $\frac{\nabla l}{\nabla m}(m, n)|_{\nabla m=h1,\ \nabla n=h2}$ of l on m, we keep n constant. Obviously, single-sided discrete left partial derivative functions and Newtonian partial derivative functions are the special cases of unified left partial derivative functions, for they do the same.

Example 21.2: Suppose $l=mn^2|_{\nabla m=3,\ \nabla n=2}$, compute $\frac{\nabla l}{\nabla m}(m, n)|_{\nabla m=3,\ \nabla n=2}$ and $\frac{\nabla l}{\nabla n}(m, n)|_{\nabla m=3,\ \nabla n=2}$.

Solution: $\frac{\nabla l}{\nabla m}(m, n)|_{\nabla m=3,\ \nabla n=2}=n^2$, $\frac{\nabla l}{\nabla n}(m, n)|_{\nabla m=3,\ \nabla n=2}=m(2n-h_2)=m(2n-2)$.

21.3.3 Unified double integrals

Suppose $l=f(m, n)|_{\nabla m=h1,\ \nabla n=h2}$, then $(\nabla m=h_1)I_{m=c}{}^{d}(\nabla n=h_2)I_{n=a}{}^{b}f(m, n)$ is unified double integral, and when integrating on *n*, we keep *m* constant. Obviously, single-sided discrete double integrals and Newtonian double integrals are the special cases of unified double integral, for they do the same.

Example 21.3: Suppose $l=mn^2|_{\nabla m=3,\ \nabla n=2}$, compute $(\nabla m=3)I_{m=0}{}^{12}(\nabla n=2)I_{n=0}{}^{8}mn^2$.

Solution:

$(\nabla m=3)I_{m=0}{}^{12}(\nabla n=2)I_{n=0}{}^{8}mn^2$

$=(\nabla m=3)I_{m=0}{}^{12}m\{2(\frac{n}{2})^3+3(\frac{n}{2})^2+(\frac{n}{2})\}/(6/2^2)|_0{}^8$

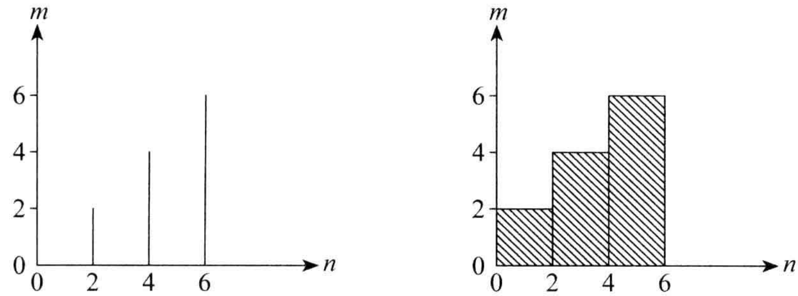
$=(\nabla m=3)I_{m=0}{}^{12}120m$

$=120\{(\frac{m}{3})^2+(\frac{m}{3})\}/(\frac{2}{2})|_0{}^{12}$

$=3600.$

21.4 Unified ordinary differential equations

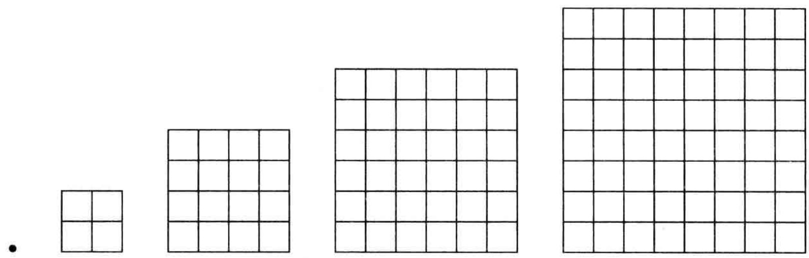
We only consider simple unified ordinary differential equations; i.e., only the highest-order derivative exists. At this time, integrals cannot be performed as $(\nabla n=h)If(n)$, but as $If(n)(\nabla n=h)$. $(\nabla n=h)I_{n=a}{}^{b}f(n)$ is different from $I_{n=a}{}^{b}f(n)(\nabla n=h)$. The geometric interpretation of the former is the sum of the length of the vertical lines enclosed by n=a+h, $f(n)$, n=b, and n-axis, and $f(n)$ is the length of any of the vertical lines. The geometric interpretation of the latter is the area of the curved trapezoid enclosed by n=a, $f(n)$, n=b, and n-axis, and $f(n)h$ is the area of any of the rectangles, $f(n)$ is the height of the rectangle, h is the width of the rectangle. For example, $(\nabla n=2)I_{n=0}{}^{6}n$ is the sum of the length of the three vertical lines shown in Fig. 21.2 (a), $(\nabla n=2)I_{n=0}{}^{6}n=2+4+6=12$. While $I_{n=2}{}^{6}n(\nabla n=2)$ is the sum of the area of the three shaded rectangles shown in Fig. 21.2 (b), the heights of the three rectangles are 2, 4, and 6 respectively, the width is 2, the area of the curved trapezoid is (2+4+6)∗2; i.e., we first compute $(\nabla n=2)I_{n=0}{}^{6}n=2+4+6$, and then we multiply it by 2.



(a) $(\nabla n=2)I_{n=0}^{6}n$ (b) $I_{n=2}^{6}n(\nabla n=2)$

Fig. 21.2 Different geometric interpretations of $(\nabla n=2)I_{n=0}^{6}n$ and $I_{n=2}^{6}n(\nabla n=2)$

Example 21.4: The squares with sides being nonnegative even numbers are shown in Fig. 21.3.



n	0	2	4	6	8
$m\|_{\nabla n=2}=n^2$	0	4	16	36	64
$\frac{\nabla m}{\nabla n}\|_{\nabla n=2}$		$\frac{4-0}{2}=2$	$\frac{16-4}{2}=6$	$\frac{36-16}{2}=10$	$\frac{64-36}{2}=14$
$\nabla^2 m/\nabla n^2\|_{\nabla n=2}$			$\frac{6-2}{2}=2$	$\frac{10-6}{2}=2$	$\frac{14-10}{2}=2$

Fig. 21.3 An example of unified ordinary differential equations

The first row are the sides of the squares, the second row are the areas of the squares, the third row are the first-order left derivatives of area to side, the fourth row are the second-order left derivatives of area to side. The initial value problem is

$$\nabla^2 m/\nabla n^2|_{\nabla n=2}=2 \tag{21.23}$$

$$m|_{n=0,\ \nabla n=2}=0 \tag{21.24}$$

$$\frac{\nabla m}{\nabla n}\Big|_{n=2,\ \nabla n=2}=2 \tag{21.25}$$

Solution: Integrating on n to both sides of (21.23), obtaining

$$I(\nabla^2 m/\nabla n^2)(\nabla n=2)=I2(\nabla n=2)$$

and

$$\frac{\nabla m}{\nabla n}\Big|_{\nabla n=2}=2\left(\frac{n}{2}\right)2+C_1=2n+C_1 \tag{21.26}$$

Substituting (21.25) into (21.26), obtaining

$C_1=-2$

and

$$\frac{\nabla m}{\nabla n}\Big|_{\nabla n=2}=2n-2 \tag{21.27}$$

Integrating on n to both sides of (21.27), obtaining

$$I\frac{\nabla m}{\nabla n}(\nabla n=2)=I(2n-2)(\nabla n=2)$$

and

$$m|_{\nabla n=2}=2\{n^2/4+n/2\}/(2/2)2-2\left(\frac{n}{2}\right)2+C_2=n^2+C_2 \tag{21.28}$$

Substituting (21.24) into (21.28), obtaining

$C_2=0$

and

$m|_{\nabla n=2}=n^2$.

Single-sided discrete ordinary differential equations are special cases of unified ordinary differential equations, because $(\nabla n=1)If(n)$ and $If(n)(\nabla n=1)$ are the same in value. For example, $(\nabla n=1)I_{n=0}^{3}n$ is the sum of the length of the three vertical lines shown in Fig. 21.4 (a), $(\nabla n=1)I_{n=0}^{3}n=1+2+3=6$. $I_{n=0}^{3}n(\nabla n=1)$ is the sum of the area of the three shaded rectangles, the heights of the rectangles are 1, 2, and 3 respectively, the width is 1, $I_{n=0}^{3}n(\nabla n=1)=(1+2+3)*1=6$.

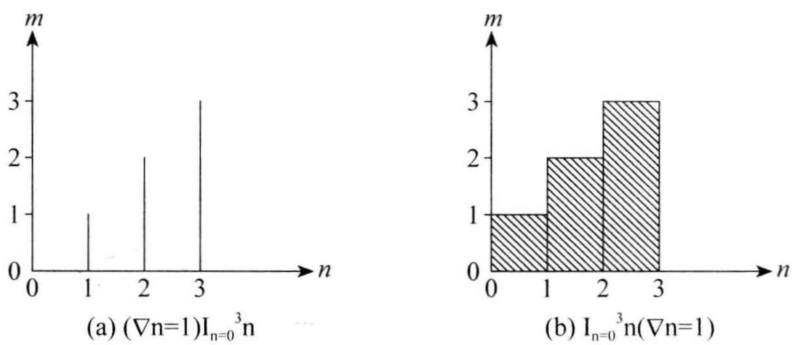


Fig. 21.4 $(\nabla n=1)I_{n=0}^{3}n$ and $I_{n=0}^{3}n(\nabla n=1)$ are the same in value

Example 21.5: Let us redo Example 20.10 using unified ordinary differential equation. The initial value problem is

$$\nabla^2 m/\nabla n^2=2 \tag{21.29}$$

$$m|_{n=0}=0 \tag{21.30}$$

$$\frac{\nabla m}{\nabla n}\Big|_{n=1,\ \nabla n=1}=1 \tag{21.31}$$

Solution: Integrate on n to both sides of (21.29), obtaining

$$I(\nabla^2 m/\nabla n^2)(\nabla n=1)=I2(\nabla n=1)$$

and

$$\frac{\nabla m}{\nabla n}=2n*1+C_1 \qquad (21.32)$$

Substitute (21.31) into (21.32), obtaining

$$C_1=-1$$

and

$$\frac{\nabla m}{\nabla n}=2n-1 \qquad (21.33)$$

Integrate on n to both sides of (21.33), obtaining

$$I\frac{\nabla m}{\nabla n}(\nabla n=1)=I(2n-1)(\nabla n=1)$$

and

$$m=2*\{n^2+n\}/2*1-n*1+C_2=n^2+C_2 \qquad (21.34)$$

Substitute (21.30) into (21.34), obtaining

$$C_2=0$$

and

$$m=n^2.$$

From this example we see that single-sided discrete ordinary differential equations are special cases of unified ordinary differential equations. Continuous ordinary differential equations are also special cases of unified ordinary differential equations, see Example 21.6.

Example 21.6: The continuous ordinary differential equation is

$$d^2y/dx^2=2 \qquad (21.35)$$

The initial condition one is

$$y|_{x=0}=0 \qquad (21.36)$$

Because in Example 21.4. when $\nabla n=2$, $\frac{\nabla m}{\nabla n}\Big|_{n=2,\ \nabla n=2}=2$; in Example 21.5, when $\nabla n=1$, $\frac{\nabla m}{\nabla n}\Big|_{n=1,\ \nabla n=1}=1$. We believe that when $\nabla n\to 0$, there is the initial condition two

$$\frac{dy}{dx}\Big|_{x=0}=0 \qquad (21.37)$$

Solution: Integrate on x to both sides of (21.35), obtaining

$$\int y''dx=\int 2dx$$

and

$$\frac{dy}{dx}=2x+C_1 \qquad (21.38)$$

Substitute (21.37) into (21.38), obtaining

$$C_1=0$$

and

$$\frac{dy}{dx}=2x \qquad (21.39)$$

Integrate on x to both sides of (21.39), obtaining

$$\int y''dx=\int 2xdx$$

and

$$y=x^2+C_2 \qquad (21.40)$$

Substitute (21.36) into (21.40), obtaining

$$C_2=0$$

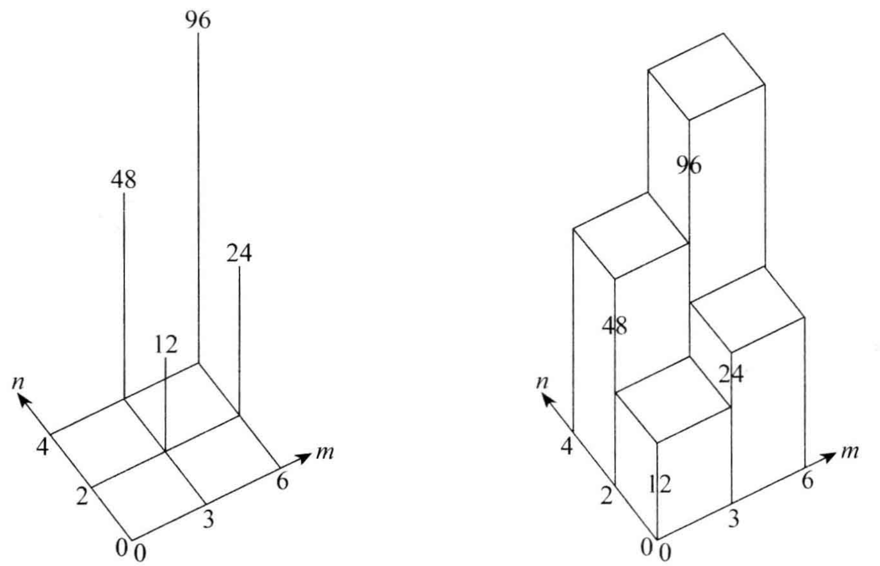
and

$$y=x^2.$$

From this example we see that continuous ordinary differential equations are special cases of unified ordinary differential equations.

21.3 Unified partial differential equations

We only consider the simple unified partial differential equations; i.e., only the highest-order derivative exists. At this time, integrals cannot be performed as $(\nabla m=h_1)I(\nabla n=h_2)If(m, n)$, but as $IIf(m, n)(\nabla n=h_2)(\nabla m=h_1)$. $(\nabla m=h_1)I_{m=c}^{d}(\nabla n=h_2)I_{n=a}^{b}f(m, n)$ is different from $I_{m=c}^{d}I_{n=a}^{b}f(m, n)(\nabla n=h_2)(\nabla m=h_1)$. The geometric interpretation of the former is the sum of length of the vertical lines in the domain $a+h_1\leqslant n\leqslant b\wedge c+h_2\leqslant m\leqslant d$, and $f(m, n)$ is the length of any of the vertical lines. The geometric interpretation of the latter is the



(a) $(\nabla m=3)I_{m=0}^{6}(\nabla n=2)I_{n=0}^{4}mn^2$ (b) $I_{m=0}^{6}I_{n=0}^{4}mn^2(\nabla n=2)(\nabla m=3)$

Fig. 21.5 Different geometric interpretations of $(\nabla m=3)I_{m=0}^{6}(\nabla n=2)I_{n=0}^{4}mn^2$ and $I_{m=0}^{6}I_{n=0}^{4}mn^2(\nabla n=2)(\nabla m=3)$

sum of the volume of the cuboids in the domain $a\leqslant n\leqslant b\wedge c\leqslant m\leqslant d$, and $f(m, n)h_1h_2$ is the volume of any of the cuboids, $f(m, n)$is the height, h_1 is the width, h_2 is the depth of the cuboid. For example, $(\nabla m=3)I_{m=0}^{6}(\nabla n=2)I_{n=0}^{4}mn^2$ is the sum of the length of the four vertical lines shown in Fig. 21.5 (a), $(\nabla m=3)I_{m=0}^{6}(\nabla n=2)I_{n=0}^{4}mn^2=12+24+48+96=180$. While $I_{m=0}^{6}I_{n=0}^{4}mn^2(\nabla n=2)(\nabla m=3)$ is the sum of the volume of the four cuboids shown in Fig. 21.5 (b), $I_{m=0}^{6}I_{n=0}^{4}mn^2(\nabla n=2)(\nabla m=3)=12*2*3+24*2*3+48*2*3+96*2*3=1080$. $I_{m=0}^{6}I_{n=0}^{4}mn^2(\nabla n=2)(\nabla m=3)$ is computed as follows: first, $(\nabla n=2)I_{n=0}^{4}mn^2$ is computed, and $20m$ is obtained. Secondly, $20m$ is multiplied by $\nabla n=2$, and $40m$ is obtained. Thirdly, $(\nabla m=3)I_{m=0}^{6}40m$ is computed, and 360 is obtained. Lastly, 360 is multiplied by $\nabla m=3$, and 1080 is obtained.

Example 21.7: Suppose $l=mn|_{\nabla m=3,\ \nabla n=2}$ shown in Fig. 21.6.

$$
\begin{array}{c c}
 & \begin{array}{ccccc} n\ 0 & 2 & 4 & 6 & 8 \end{array} \\
m & \\
\begin{array}{c} 0 \\ 3 \\ 6 \\ 9 \\ 12 \end{array} &
\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 12 & 18 & 24 \\ 0 & 12 & 24 & 36 & 48 \\ 0 & 18 & 36 & 54 & 72 \\ 0 & 24 & 48 & 72 & 96 \end{bmatrix}
\end{array}
$$

Fig. 21.6 Matrix representation of $l=mn|_{\nabla m=3,\ \nabla n=2}$

We have $\frac{\nabla l}{\nabla m}(m, n)|_{\nabla m=3,\ \nabla n=2}=n$, $\nabla^2 l/\nabla n\nabla m(m, n)|_{\nabla m=3,\ \nabla n=2}=1$. The initial value problem is

$$\nabla^2 l/\nabla n\nabla m(m, n)|_{\nabla m=3,\ \nabla n=2}=1 \tag{21.41}$$

$$l|_{m=0,\ n=0,\ \nabla m=3,\ \nabla n=2}=0 \tag{21.42}$$

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=3,\ n=2,\ \nabla m=3,\ \nabla n=2}=\frac{6-0}{3}=2 \tag{21.43}$$

Solution: Integrate on n to both sides of (21.41), obtaining

$$I\nabla^2 l/\nabla n\nabla m(\nabla n=2)=I1(\nabla n=2)$$

and

$$\frac{\nabla l}{\nabla m}|_{\nabla n=2}=\frac{n}{2}*2+C_1 \tag{21.44}$$

Substitute (21.43) into (21.44), obtaining

$$C_1=0$$

and

$$\frac{\nabla l}{\nabla m}|_{\nabla n=2}=n \tag{21.45}$$

Integrate on m to both sides of (21.45), obtaining

$$I\frac{\nabla l}{\nabla m}(\nabla m=3)=In(\nabla m=3)$$

and

$$l|_{\nabla m=3,\ \nabla n=2}=n*\frac{m}{3}*3+C_2 \tag{21.46}$$

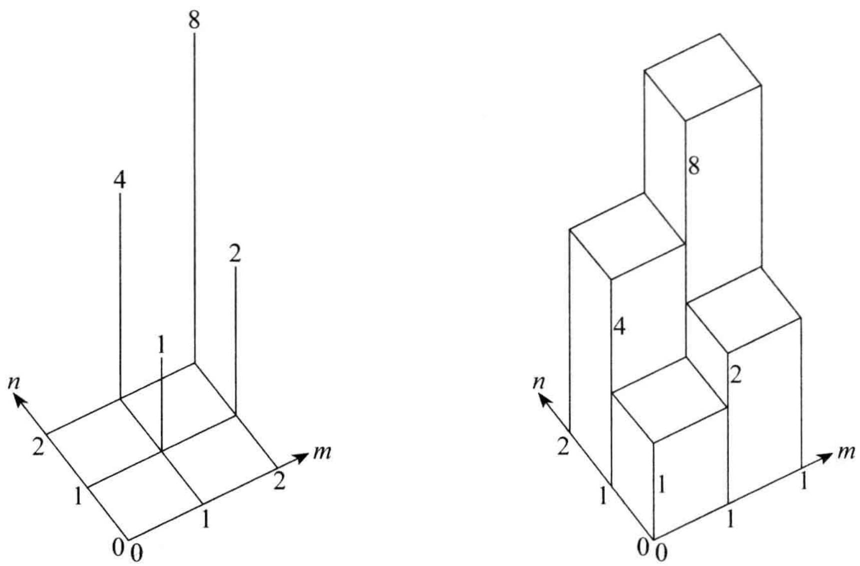
Substitute (21.42) into (21.46), obtaining

C2=0

and

$l|_{\nabla m=3,\ \nabla n=2}=mn.$

Single-sided discrete partial differential equations are special cases of unified partial differential equations, because $(\nabla m=1)I(\nabla n=1)If(m, n)$ and $IIf(m, n)(\nabla n=1)(\nabla m=1)$ are the same in value. For example, $(\nabla m=1)I_{m=0}^{2}(\nabla n=1)I_{n=0}^{2}mn^2$ is the sum of the length of the four vertical lines shown in Fig. 21.7 (a), $(\nabla m=1)I_{m=0}^{2}(\nabla n=1)I_{n=0}^{2}mn^2=1+2+4+8=15$. $I_{m=0}^{2}I_{n=0}^{2}mn^2(\nabla n=1)(\nabla m=1)$ is the sum of the volume of the four cuboids shown in Fig. 21.7 (b), $I_{m=0}^{2}I_{n=0}^{2}mn^2(\nabla n=1)(\nabla m=1)=1*1*1+2*1*1+4*1*1+8v1*1=15$.



(a) $(\nabla m=1)I_{m=0}^{2}(\nabla n=1)I_{n=0}^{2}mn^2$ (b) $I_{m=0}^{2}I_{n=0}^{2}mn^2(\nabla n=1)(\nabla m=1)$

Fig. 21.7 $(\nabla m=1)I_{m=0}^{2}(\nabla n=1)I_{n=0}^{2}mn^2$ and $I_{m=0}^{2}I_{n=0}^{2}mn^2(\nabla n=1)(\nabla m=1)$ are the same in value

Example 21.8: We redo Example 20.11 using unified partial differential equations.

Solution: The initial value problem is

$$\nabla^2 l/\nabla m\nabla n(m, n)=1 \tag{21.47}$$

$$l|_{m=0,\ n=0}=0 \tag{21.48}$$

$$\frac{\nabla l}{\nabla m}(m, n)|_{m=1,\ n=1}=1 \tag{21.49}$$

Integrate on n to both sides of (21.47), obtaining

$$\mathrm{I}\nabla^2 l/\nabla m\nabla n(\nabla n=1)=\mathrm{I}1(\nabla n=1)$$

and

$$\frac{\nabla l}{\nabla m}(m,\ n)=n+C_1 \tag{21.50}$$

Substitute (21.49) into (21.50), obtaining

$$C_1=0$$

and

$$\frac{\nabla l}{\nabla m}(m,\ n)=n \tag{21.51}$$

Integrate on m to both sides of (21.51), obtaining

$$\mathrm{I}\frac{\nabla l}{\nabla m}(m,\ n)(\nabla m=1)=\mathrm{I}n(\nabla m=1)$$

and

$$l=mn+C_2 \tag{21.52}$$

Substitute (21.48) into (21.52), obtaining

$$C_2=0$$

and

$$l=mn.$$

From this example we see that single-sided discrete partial differential equations are special cases of unified partial differential equations.

Continuous partial differential equations are also special cases of unified partial differential equations, see Example 21.9.

Example 21.9: The continuous partial differential equation is

$$\partial^2 z/\partial y\partial x=1 \tag{21.53}$$

The initial condition one is

$$z|_{x=0,\ y=0}=0 \tag{21.54}$$

In Example 21.7, when $\nabla n=2$, $\frac{\nabla l}{\nabla m}(m,\ n)|_{m=3,\ n=2,\ \nabla m=3,\ \nabla n=2}=2$. In Example 21.8, when $\nabla n=1$, $\frac{\nabla l}{\nabla m}(m,\ n)|_{m=1,\ n=1,\ \nabla m=1,\ \nabla n=1}=1$. We believe that when $\nabla n\to 0$, we have the initial condition two

$$\frac{\partial z}{\partial y}|_{x=0,\ y=0}=0 \tag{21.55}$$

Solution: Integrate on x to both sides of (21.53), obtaining

$$\int\partial^2 z/\partial y\partial x dx=\int 1dx$$

and

$$\frac{\partial z}{\partial y}=x+C_1 \tag{21.56}$$

Substitute (21.55) into (21.56), obtaining

$$C_1=0$$

and

$$\frac{\partial z}{\partial y}=x \tag{21.57}$$

Integrate on *y* to both sides of (21.57), obtaining

$$\int\frac{\partial z}{\partial y}dy=\int xdy$$

and

$$z=xy+C_2 \tag{21.58}$$

Substitute (21.54) into (21.58), obtaining

$$C_2=0$$

and

$$z=xy.$$

From this example, we see that continuous partial differential equations are special cases of unified partial differential equations.

Part 9
Mutually-inversistic abstract algebra

Mutually-inversistic abstract algebra inherits from classical abstract algebra and has many innovations. For example, in mutually-inversistic abstract algebra, new properties such as binary bijection, complement idempotency are proposed; new algebras such as Boolean sum-Boolean product algebra, Nand-Nor algebra, main-auxiliary algebras of set theorems are constructed; transaxis (transaxis is a new English word coined by the author) straight lines in mutually inverse diagrams are used to denote double-sided discrete linear partial derivative functions, and to reveal algebraic properties. Algebraic systems are reclassified. Vertically, according as whether an algebra is described by a term space or a fact space, the algebra is classified as algebra of term space or algebra of fact space. Horizontally, according as whether the operators of an algebra contain the auxiliary only or contain both the auxiliary and the partially ordered main, the algebra is classified as auxiliary algebra or main-auxiliary algebra. Auxiliary algebras are further classified into associative auxiliary algebras (including semigroup and monoid), binary bijective auxiliary algebra (including group, ring, field, Boolean sum-Boolean product algebra), idempotent auxiliary algebra (including lattice), complementary idempotent auxiliary algebra (including NAND-NOR algebra). Main-auxiliary algebras are further classified into lattice, Boolean algebra, main-auxiliary algebras of set theorems. In Chapter 22, auxiliary algebras are studied. In Chapter 23, main-auxiliary algebras are studied.

Chapter 22 Auxiliary algebras

22.1 Algebraic structures

22.1.1 Constituents of an algebra

An algebra is usually composed of three parts:

(1) A carrying set.

(2) Operations defined on the carrying set. Operations can be unary or binary. Operations should be closed; i.e., the results of the operations should be within the carrying set.

(3) Distinguished elements of the carrying set, called algebraic constants. Some algebras do not contain algebraic constants.

An algebra is usually denoted by an n-tuple of the carrying set, the operations, and the algebraic constants.

Example 22.1: (a) Integers, addition, and constant 0 constitute an algebra of term space.

(1) The carrying set is $\mathbf{Z}=\{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

(2) The operation defined on $\mathbf{Z}$ is addition (denoted as +).

(3) The constant is 0.

(4) This algebra is denoted as $\langle \mathbf{Z}, +, 0\rangle$.

(b) The power set ρ(S), ∪, ∩, ~, ∅, and S constitute an algebra of fact space.

(1) The carrying set is the power set ρ(S) of S.

(2) The operations defined on ρ(S) are ∪, ∩, ~.

(3) The constants are ∅ and S.

This algebra is denoted as $\langle \rho(S), \cup, \cap, \sim, \emptyset, S\rangle$.

22.1.2 Identities and zero elements

Identities and zero elements are algebraic constants.

Definition 22.1: Suppose $*$ is a binary operation on S (S is a subset of I_0, which is a domain of terms; i.e., the set of zeroth-order terms), 1_l is an element of S. If for every element x in S, we have

$1_l * x = x$

then we say that 1_l is the left identity of term space for the operation $*$. Suppose 0_l is an element of S. If for every element x in S, we have

$0_l * x = 0_l$

then we say 0_l is the left zero element of term space for the operation $*$.

Similarly, we can define the right identity of term space 1_r and right zero element of term space 0_r.

Definition 22.2: Suppose $*$ is an operation on S ($S \subseteq {}^{-1}I_0$), 1 is an element in S. If for every element x in S, we have

$1 * x = x * 1 = x$

then we say that 1 is the identity of term space for the operation $*$. Suppose 0 is an element of S. If for every element x in S, we have

$0 * x = x * 0 = 0$

then we say that 0 is the zero element of term space for the operation $*$.

Definition 22.3: Suppose $\odot$ is a binary operation on S (S is a subset of I_1, which is the domain of fact propositions; i.e., the set of first-order fact propositions), $\mathbf{1}_l$ is an element of S. If for every element P in S, we have

$\mathbf{1}_l \odot P = {}^{-1} P$

then we say that $\mathbf{1}_l$ is the left identity of fact space for the operation $\odot$. Suppose $\mathbf{0}_l$ is an element in S. If for every element P in S, we have

$\mathbf{0}_l \odot P = {}^{-1} \mathbf{0}_l$

then we say that $\mathbf{0}_l$ is the left zero element of fact space for the operation $\odot$.

Similarly, we can define right identity of fact space $\mathbf{1}_r$ and right zero element of fact space $\mathbf{0}_r$.

Definition 22.4: Suppose $\odot$ is a binary operation on S ($S \subseteq {}^{-1}I_1$), $\mathbf{1}$ is an element in S. If for every element P in S, we have

$\mathbf{1} \odot P = {}^{-1} P \odot \mathbf{1} = {}^{-1} P$

then we say that $\mathbf{1}$ is the identity of fact space for the operation $\odot$. Suppose $\mathbf{0}$ is an element in S. If for every element P in S, we have

$\mathbf{0} \odot P = {}^{-1} P \odot \mathbf{0} = {}^{-1} \mathbf{0}$

then we say that $\mathbf{0}$ is the zero element of fact space for the operation $\odot$.

Example 22.2: The algebra of term space ⟨{a, b, c}, $*$⟩ is defined by Table 22.1.

Table 22.1 Operation table for ⟨{a, b, c}, $*$⟩

$*$	a	b	c
a	a	b	b
b	a	b	c
c	a	b	a

From Table 22.1 we learn that a and b are the right zero elements of term space, there

is no left zero element of term space; b is the left identity of term space, there is no right identity of term space.

Example 22.3: (1) The algebra of term space <**Z**, •, 1, 0>, where • denotes multiplication, has an identity 1 and a zero element 0.

(2) The algebra of term space <**N**, +> has an identity 0 but no zero element.

(3) Suppose S is a finite set, the algebra of fact space <ρ(S), ∪, ∩, ~, ∅, S> has two binary operations. For the operation ∪, ∅ is the identity, S is the zero element; for the operation ∩, ∅ is the zero element, S is the identity.

22.1.3 Inverse elements

If an algebra has an identity, then we can define inverse elements.

Definition 22.5: Suppose ∗ is a binary operation on S ($S\subseteq^{-1}I_0$),1 is the identity for ∗. If $x*y=1$, then we say that x is the left inverse element of term space of y with respect to ∗, and y is the right one of x. If both $x*y=1$ and $y*x=1$ hold, then we say that x is the inverse element of term space of y (and y is also the inverse element of term space of x). The inverse element of x is usually denoted by x^{-1}.

Similarly, we can define the inverse element (left inverse element, right inverse element) of fact space. If the inverse element (left inverse element, right inverse element) of an element exists, then the element is said to be invertible (left invertible, right invertible).

Example 22.4: The algebra of term space <{a, b, c}, ∗> is defined as Table 22.2.

Table 22.2 Operation table for <{a, b, c}, *>

∗	a	b	c
a	a	a	b
b	a	b	c
c	a	c	c

From Table 22.2 we learn that b is the identity, the right inverse element of a is c, the inverse element of b is itself, the left inverse element of c is a.

Example 22.5: (1) The algebra of term space <**Z**, +> has the identity 0, every element $x\in\mathbf{Z}$ has an inverse element $-x$ with respect to +, for $x+(-x)=0$.

(2) For the algebra of term space <**N**, +>, only the identity 0 has an inverse element, which is itself.

(3) For the algebra of term space <**R**, •>, all elements but the zero element 0 have inverse elements.

(4) For the algebra of fact space <ρ(S), ∪>, only the identity ∅ has inverse element which is itself.

22.1.4 Complementary identities, complementary zero elements, and complementary inverse elements

Definition 22.6: Suppose $\odot$ is a binary operation on $S(S\subseteq^{-1}I_l)$, $\widetilde{\mathbf{1}}_l$ is an element of S. If for every element P in S, we have

$$\widetilde{\mathbf{1}}_l\odot P=^{-1}\sim P$$

then we say that $\widetilde{\mathbf{1}}_l$ is the left complementary identity of fact space with regard to $\odot$. Suppose $\widetilde{\mathbf{0}}_l$ is an element of S. If for every element P in S, we have

$$\widetilde{\mathbf{0}}_l\odot P=^{-1}\sim\widetilde{\mathbf{0}}_l$$

then we say that $\widetilde{\mathbf{0}}_l$ is the left complementary zero element of fact space with regard to $\odot$.

Similarly, we can define right complementary identity of fact space and right complementary zero element of fact space.

Definition 22.7: Suppose $\odot$ is a binary operation on $S(S\subseteq^{-1}I_l)$, $\widetilde{\mathbf{1}}$ is an element of S. If for every element P in S, we have

$$\widetilde{\mathbf{1}}\odot P=^{-1}P\odot\widetilde{\mathbf{1}}=^{-1}\sim P$$

then we say that $\widetilde{\mathbf{1}}$ is the left complementary identity of fact space with regard to $\odot$. Suppose $\widetilde{\mathbf{0}}$ is an element of S. If for every element P in S, we have

$$\widetilde{\mathbf{0}}\odot P=^{-1}P\odot\widetilde{\mathbf{0}}=^{-1}\sim\widetilde{\mathbf{0}}$$

then we say that $\widetilde{\mathbf{0}}$ is the left complementary zero element of fact space with regard to $\odot$.

If the complementary identity in an algebra exists, then the complementary inverse element can be defined.

Definition 22.8: Suppose $\odot$ is a binary operation on $S(S\subseteq^{-1}I_l)$, $\widetilde{\mathbf{1}}$ is the complementary identity of fact space with regard to $\odot$. If for every elements P, Q in S, we have

$$P\odot Q=^{-1}\widetilde{\mathbf{1}}$$

then we say that P is the left complementary inverse element of fact space of Q and Q is the right complementary inverse element of fact space of P with regard to $\odot$. If we have

$$P\odot Q=^{-1}Q\odot P=^{-1}\widetilde{\mathbf{1}}$$

then we say that P is the complementary inverse element of fact space of Q (of course, Q is also the complementary inverse element of fact space of P). The complementary inverse element of fact space of P is denoted as $P^{\sim-1}$.

An element having a complementary inverse element (left complementary inverse element, right complementary inverse element) is said to be complementary invertible (left complementary invertible, right complementary invertible).

22.2 Transaxis straight lines

Definition 22.9: If there exists at least one axis, the projection of a straight line on the

axis span over the whole axis, then the straight line is called a transaxis straight line.

Example 22.6: The mutually inverse diagram of $l=n+_4m$ ($+_4$ is addition modulo 4) is shown in Fig. 22.1. In Fig. 22.1, straight line $A_1A_2A_3A_4$ is a transaxis straight line, because, for the n-axis, the projection of A_1 is on n=0, that of A_2 is on n=1, that of A_3 is on n=2, that of A_4 is on n=3. For the same reason, the projection of $A_1A_2A_3A_4$ is on the whole l-axis. While straight line D is not a transaxis straight line, because for the n-axis, no projection is on n=0; for the m-axis, no projection is on m=0; for the l-axis, no projections are on l=1, 2, 3.

In Chapter 19, every straight line denotes a double-sided discrete derivative. In this chapter, every transaxis straight line denotes a derivative, reveals an algebraic property. While non-transaxis straight lines do not denote derivatives, do not reveal algebraic properties.

Example 22.7: In Fig. 22.1, the transaxis straight line $A_1A_2A_3A_4$ denotes the partial derivative $\frac{\partial l}{\partial n}\Big|_{m=0}=1$. It reveals that as n increases, l increases linearly; and that when m=0,

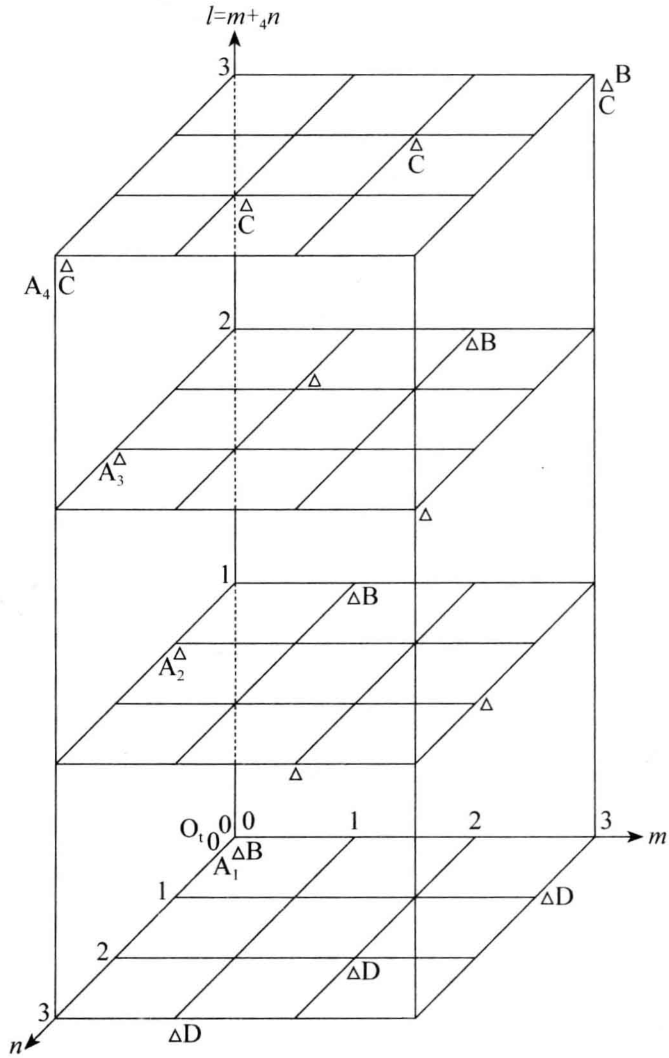


Fig. 22.1 Transaxis straight lines

$n+_4 0=n$; i.e., $m=0$ is the right identity with regard to $+_4$. Likewise, the transaxis straight line B denotes $\frac{\partial l}{\partial m}|_{n=0}=1$, reveals that $n=0$ is the left identity with regard to $+_4$. The transaxis straight line C denotes $\frac{\partial m}{\partial n}|_{l=3}=-1$. It reveals that when $l=3$, as n increases, m decreases linearly, their sum retains the maximum value 3. The non-transaxis straight line D does not denote a derivative, does not reveal an algebraic property.

22.3 Associative auxiliary algebras

Associative auxiliary algebras include semigroup and monoid.

Definition 22.10: An algebra having the constituent ⟨S, $*$⟩, where $*$ is a binary operation, and satisfying the associative law

$x*(y*z)=(x*y)*z$

is called a semigroup of term space.

Similarly, we can define semigroup of fact space.

Definition 22.11: An algebra having the constituent ⟨S, $*$, 1⟩, where $*$ is a binary operation, 1 is the identity, and satisfying the associative law

$x*(y*z)=(x*y)*z$

is called a monoid of term space.

Similarly, we can define monoid of fact space.

Example 22.8: (1) The algebra ⟨**R**, •⟩ is a semigroup of term space. The algebra ⟨**R**, •, 1⟩ is a monoid of term space.

(2) The algebras ⟨$\mathbf{N}_k$, $+_k$, 0⟩ and ⟨$\mathbf{N}_k$, $\times_k$, 1⟩ are both monoids of term space.

(3) The algebras ⟨ρ(S), ∪, ∅⟩ and ⟨ρ(S), ∩, S⟩ are both monoids of fact space.

(4) The algebra ⟨$\mathbf{Z}_+$, +⟩, where $\mathbf{Z}_+$ is the set of positive integers, is a semigroup of term space, but not a monoid of term space.

22.4 Binary bijective auxiliary algebras

22.4.1 Groups

Definition 22.12: The group of term space ⟨G, $*$⟩($G\subseteq{}^{-1}I_0$) is an algebra, the binary operation $*$ of which satisfies the following three conditions:

(1) For all $x, y, z\in G$, $x*(y*z)=(x*y)*z$.

(2) There exists the identity e such that for any element $x\in G$, we have $x*e=e*x=x$.

(3) For any $x\in G$, there exists an inverse element x^{-1} such that $x^{-1}*x=x*x^{-1}=e$.

Similarly, we can define the group of fact space.

In a group, the inverse element of every element is unique. Therefore the inverse operation -1 can be viewed as a unary operation, and the constituent of a group can be written as <G, ∗, −1, e>.

If G is a finite set, then <G, ∗> is called a finite group. If G is an infinite set, then <G, ∗> is called an infinite group. The number of elements in a finite group G is called the order of the group.

The operation ∗ in a group is usually called multiplication. If ∗ is commutative, then the group is called a commutative group or Abelian group. In a commutative group, if the operator ∗ is changed to +, then the group is called an additive group, and the inverse element x^{-1} is written as $-x$.

Example 22.9: (1) The algebra <**Z**, +, −, 0> is an Abelian group of term space, where + denotes addition, − denotes unary subtraction.

(2) The algebra <$\mathbf{Q}_{+}$, •, −1, 1> is an Abelian group of term space, where • denotes multiplication, −1 denotes reciprocal operation.

(3) The algebra <$\mathbf{N}_4$, $+_4$, −1, 0> is an Abelian group of term space, where x^{-1}=4−x. The mutually inverse diagram of $+_4$ has been shown in Fig. 22.1. The partial derivatives of Fig. 22.1 has been given in Example 22.7. Fig. 22.1 can be crushed into an operation table shown in Table 22.3.

Table 22.3 Operation table for $+_4$

$+_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Example 22.10: Prove that <ρ(S), ⊕, −1, ∅> makes an Abelian group of fact space.

Proof: First, both $\{P \oplus Q\} \oplus R$ and $P \oplus \{Q \oplus R\}$ are denoted by the shaded compartments of Fig. 22.2, therefore the associative law

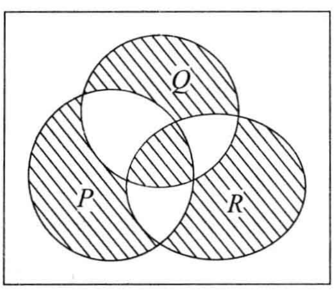


Fig. 22.2 ⊕ satisfies associativity

$\{P \oplus Q\} \oplus R =^{-1} P \oplus \{Q \oplus R\}$

holds.

Secondly, for any $P \in \rho(S)$, we have

$\varnothing \oplus P =^{-1} P \oplus \varnothing =^{-1} P$ (according to the definition of $\oplus$)

That is, $\varnothing$ is the identity. Thirdly, for any $P \in \rho(S)$, we have

$P \oplus P =^{-1} \varnothing$ (according to the definition of $\oplus$)

That is, the inverse element of P is itself: $P^{-1} =^{-1} P$. Therefore, $\langle \rho(S), \oplus, -1, \varnothing \rangle$ makes a group of fact space. Lastly, both $P \oplus Q$ and $Q \oplus P$ are denoted by the shaded compartments of Fig. 22.3, therefore, the commutative law

$\{P \oplus Q\} =^{-1} \{Q \oplus P\}$

holds. Therefore, $\langle \rho(S), \oplus, -1, \varnothing \rangle$ makes an Abelian group of fact space.

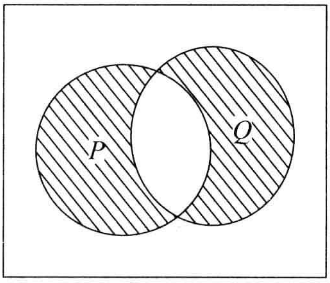


Fig. 22.3 $\oplus$ satisfies commutativity

Similarly, we can prove that $\langle \rho(S), \otimes, -1, S \rangle$ makes an Abelian group of fact space.

Example 22.11: The mutually inverse diagram of $R =^{-1} P \oplus Q$ is shown in Fig. 22.4. In Fig. 22.4, the transaxis straight line A denotes the partial derivative $\frac{\partial R}{\partial P}\Big|_{Q=^{-1}\varnothing}=1$, reveals that $Q =^{-1} \varnothing$ is the right identity with regard to $\oplus$; the transaxis straight line B denotes the partial derivative $\frac{\partial R}{\partial Q}\Big|_{P=^{-1}\varnothing}=1$, reveals that $P =^{-1} \varnothing$ is the left identity with regard to $\oplus$; the transaxis straight line C denotes the partial derivative $\frac{\partial Q}{\partial P}\Big|_{R=^{-1}\varnothing}=1$, reveals that $P \oplus P =^{-1} \varnothing$ (where $P =^{-1} Q$); i.e., the inverse element of P is itself; the transaxis straight line D denotes the partial derivative $\frac{\partial Q}{\partial P}\Big|_{R=^{-1}S}=-1$, reveals the law of excluded middle and non-contradiction $P \oplus \sim P =^{-1} S$ (where $\sim P =^{-1} Q$). The mutually inverse diagram of $Q =^{-1} P^{-1}$ in $\langle \rho(S), \oplus, -1, \varnothing \rangle$ is shown in Fig. 22.5. The transaxis straight line in Fig. 22.5 denotes derivative $\frac{dQ}{dP}=1$, reveals $P =^{-1} P^{-1}$ (where $P =^{-1} Q$).

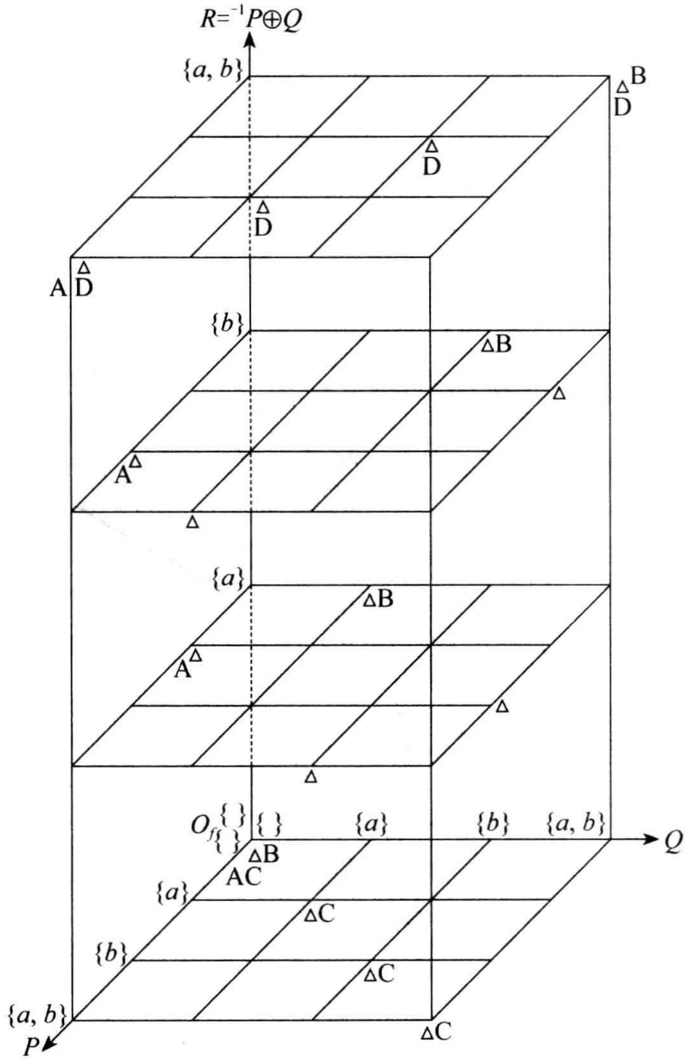


Fig. 22.4 $R={}^{-1}P\oplus Q$

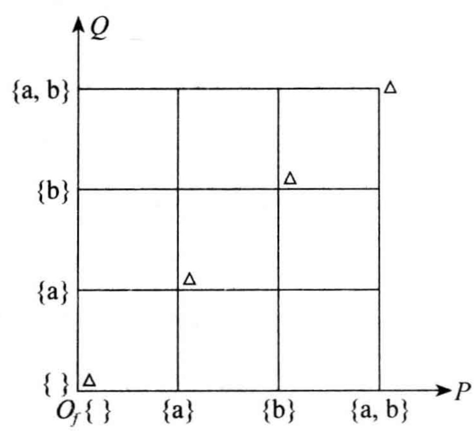


Fig.22.5 $Q={}^{-1}P^{-1}$

Groups are special cases of monoids, which are special cases of semigroups.

Theorem 22.1: Identity is the only idempotent element in a group.

Proof: If x is an idempotent element, then

$e=x^{-1}*x=x^{-1}*(x*x)=(x^{-1}*x)*x=e*x=x.$

Q.E.D.

Definition 22.13: Suppose $*$ is a binary operation on S, a$\in$S. If for every x, $y\in$S, we have

$(a*x=a*y)\vee(x*a=y*a)\leqslant^{-1}(x=y)$

then we say that a is eliminable.

Theorem 22.2: Suppose $*$ is an associative operation on S. If a$\in$S is invertible, then a is eliminable.

Proof: Suppose x, $y\in$S and $a*x=a*y$ (the proof of $x*a=y*a$ is similar, so omitted), because $*$ is associative and a is invertible (the inverse element of a is denoted by a^{-1}), thus

$a^{-1}*(a*x)=(a^{-1}*a)*x=x,$

$a^{-1}*(a*y)=(a^{-1}*a)*y=y.$

Considering

$a^{-1}*(a*x)=a^{-1}*(a*y).$

Therefore, $x=y$; i.e., a is eliminable.

Q.E.D.

Theorem 22.3: Suppose $\langle G, *\rangle$ is a group, then for any $x, y, z\in G$,

$x*y=x*z\leqslant^{-1}y=z,$

$y*x=z*x\leqslant^{-1}y=z.$

Proof: Because every element in a group has an inverse element, according to Theorem 22.2, this theorem obviously holds.

Q.E.D.

Theorem 22.4: Every row and column of the operation table of a group $\langle G, *\rangle$ are permutations (i.e., unary bijections) of the elements in G.

Proof: First, we prove that the occurrence of an element in a row or column of the operation table cannot be more than once. We use proof by contradiction. Suppose that two elements in the row corresponding to element a$\in$G are all k; i.e., $a*b_1=a*b_2=k$, and $b_1\neq b_2$. But according to Theorem 22.3, we have $b_1=b_2$. So, contradiction occurs. The same applies to columns.

Secondly, we prove that every element in G occurs in every row and every column of the operation table. Consider the row corresponding to element a. Suppose b is any element in G. Because $b=a*(a^{-1}*b)$, b ought to occur in the row corresponding to a. The same applies to columns.

Lastly, because $\langle G, *\rangle$ contains the identity, no two rows and columns are the same.

To sum up: every row of the operation table is a permutation of the elements in G, and every row is a different permutation. The same applies to columns.

Q.E.D.

Theorem 22.4 concerns operation tables, which are crushed from mutually inverse diagrams. Restore the operation table of Theorem 22.4 to mutually inverse diagram, restore the rows and columns of the operation table to sections perpendicular to the x-axis and y-axis, restore the permutations to unary bijections, then Theorem 22.4 can be restated as Theorem 22.5:

Theorem 22.5: In the mutually inverse diagram of the binary operation $*$ of the group $\langle G, *\rangle$, every section perpendicular to the x-axis is a unary bijection, and every section perpendicular to the y-axis is also a unary bijection; i.e., the mutually inverse diagram of $*$ is a binary bijection.

Theorem 22.5 tells us that a group is a binary bijective auxiliary algebra with one binary operation.

Example 22.12: (1) $\langle \mathbf{N}_4, +_4, {}^{-1}, 0\rangle$ is a group of term space. The mutually inverse diagram of $+_4$ is shown in Fig. 22.1, which can be decided to be a binary bijection.

(2) $\langle \rho\{a, b\}, \oplus, {}^{-1}, \varnothing\rangle$ is a group of fact space. The mutually inverse diagram of $\oplus$ is shown in Fig. 22.4, which can be decided to be a binary bijection.

There is only one order one group, whose operation table is shown in Table 22.4.

Table 22.4 Operation table for order one group

$*$	e
e	e

Idempotent auxiliary algebras are auxiliary algebras satisfying idempotent law and commutative law. From Theorem 22.1 we know that order one group is the only group that satisfies idempotent law. From Table 22.4 we know that order one group satisfies commutative law. Therefore order one group is the only binary bijective auxiliary algebra that is also an idempotent auxiliary algebra. Complementary idempotent auxiliary algebras are auxiliary algebras satisfying complementary idempotent law and commutative law. If we stipulate that the complement e' of the element e in the order one group as e'=e, then we have

$e*e=e=e'$.

That is, order one group satisfies complementary idempotent law. From Table 22.4 we know that order one group satisfies commutative law. Therefore, order one group is also a complementary idempotent auxiliary algebra.

There is only one order two group, whose operation table is shown in Table 22.5.

Table 22.5 Operation table for order two group

*	e	a
e	e	a
a	a	e

There is only one order three group, whose operation table is shown in Table 22.6.

Table 22.6 Operation table for order three group

*	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

If we stipulate e'=e, a'=b, b'=a, then the mutually inverse diagram of the order three group is shown in Fig. 22.6.

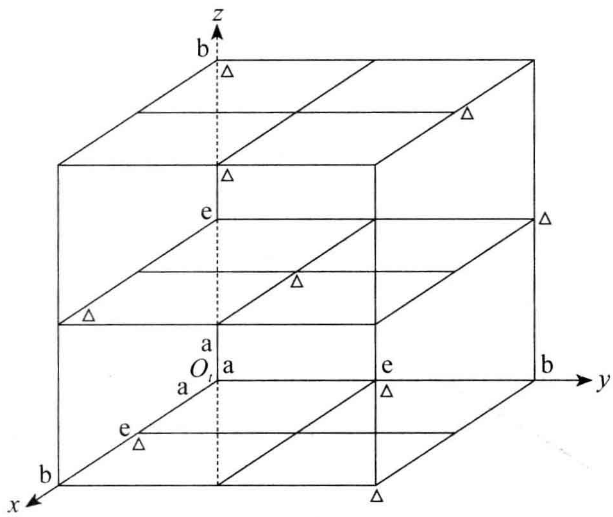


Fig.22.6 Order three group and complementary idempotent auxiliary algebra

From Fig. 22.6 we know that order three group satisfies complementary idempotent law (the vertices on the middle auxiliary diagonal axis belong to the mutually inverse diagram) and commutative law (the mutually inverse diagram is symmetric with regard to the main diagonal plain). Therefore, it is also a complementary idempotent auxiliary algebra.

There are only two order four groups, whose operation tables are shown in Tables 22.7 and 22.8.

Table 22.7 Operation table one for order four group

*	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

Table 22.8 Operation table two for order four group

*	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

There is only one order five group, whose operation table is shown in Table 22.9.

Table 22.9 Operation table for order five group

*	e	a	b	c	d
e	e	a	b	c	d
a	a	b	c	d	e
b	b	c	d	e	a
c	c	d	e	a	b
d	d	e	a	b	c

There are only two order six groups, whose operation tables are shown in Tables 22.10 and 22.11.

Table 22.10 Operation table one for order six group

*	e	a	b	c	d	f
e	e	a	b	c	d	f
a	a	b	c	d	f	e
b	b	c	d	f	e	a
c	c	d	f	e	a	b
d	d	f	e	a	b	c
f	f	e	a	b	c	d

Table 22.11 Operation table two for order six group

*	e	a	b	c	d	f
e	e	a	b	c	d	f
a	a	e	d	f	b	c
b	b	f	e	d	c	a
c	c	d	f	e	a	b
d	d	c	a	b	f	e
f	f	b	c	a	e	d

22.4.2 Rings and fields

Rings and fields are binary bijective auxiliary algebras with two binary operations.

Definition 22.14: If the binary operations + and • of the algebra ⟨R, +, •⟩ satisfy the following three properties:

(1) ⟨R, +⟩ is an Abelian group.

(2) ⟨R, •⟩ is a semigroup.

(3) Multiplication • is distributive over addition +; i.e., for any elements x, y, $z \in$R, we have

$x \bullet (y+z)=x \bullet y+x \bullet z$

$(y+z) \bullet x=y \bullet x+z \bullet x$

then we say that ⟨R, +, •⟩ is a ring of term space.

The third property in the definition is to link • and +.

Similarly, we can define rings of fact space.

Example 22.13: (1) The algebra ⟨**Z**, +, •⟩ is a ring of term space, because ⟨**Z**, +⟩ is an additive group, ⟨**Z**, •⟩ is a semigroup, • is distributive over +.

(2) The algebra $\langle \mathbf{N}_k, +_k, \times_k \rangle$ is a ring of term space.

(3) $\langle M_n, +, \bullet \rangle$ is a ring of term space. Here, M_n is a set of n×n planar matrices on **Z**, + is the matrix addition, • is the matrix multiplication. $\langle M_n, + \rangle$ is an Abelian group, the matrix of zeros is the identity, $\langle M_n, \bullet \rangle$ is a semigroup, matrix multiplication is distributive over matrix addition.

(4) The algebra $\langle \{[0]_4, [1]_4, [2]_4, [3]_4\}, +_4, \times_4 \rangle$ is a ring of fact space.

Theorem 22.6: Suppose ⟨R, +, •⟩ is a ring of term space, 0 is the additive identity, then for any element $x \in$R, we have

$x \bullet 0=0 \bullet x=0.$

That is, the additive identity is the multiplicative zero element.

Proof: $0=x \bullet 0-x \bullet 0=x \bullet (0+0)-x \bullet 0=x \bullet 0+x \bullet 0-x \bullet 0=x \bullet 0$. Similarly, we can prove $0=0 \bullet x$. (In the proof, $x-y$ is the abbreviation of $x+(-y)$).

Q.E.D.

Definition 22.15: Suppose ⟨R, +, •⟩ is a ring of term space. If there are some non-zero elements a, b∈R, such that a•b=0, then a, b are called zero divisors, ⟨R, +, •⟩ is called a ring of term space with zero divisor. A ring of term space with no zero divisor is called a ring of term space without zero divisor.

Similarly, we can define rings of fact space without zero divisor.

Definition 22.16: Given a ring of term space ⟨R, +, •⟩. If ⟨R, •⟩ is commutative, then ⟨R, +, •⟩ is called a commutative ring; if ⟨R, •⟩ is a semigroup with identity; i.e., a monoid, then ⟨R, +, •⟩ is called a ring with identity. If ⟨R, +, •⟩ is commutative, with identity, and without zero divisor, then it is called a domain of term space.

Similarly, we can define domain of fact space.

Example 22.14: (1) $\langle \mathbf{Z}, +, \bullet \rangle$ is a domain of term space.

(2) $\langle \mathbf{N}_6, +_6, \times_6 \rangle$ is not a domain of term space. Because $3\times_6 2=0$, 3 and 2 are zero divisors. $\langle \mathbf{N}_7, +_7, \times_7 \rangle$ is a domain of term space.

Definition 22.17: If $\langle F, +, \bullet \rangle$ is a domain of term space, $|F|>1$ ($|F|$ denotes the number of elements in F), $\langle F-\{0\}, \bullet \rangle$ is a group, then $\langle F, +, \bullet \rangle$ is a field of term space.

Fields of term space can also be defined as follows: If

(1) $\langle F, + \rangle$ is an Abelian group.

(2) $\langle F-\{0\}, \bullet \rangle$ is an Abelian group.

(3) $\bullet$ is distributive over $+$.

Then $\langle F, +, \bullet \rangle$ is called a field of term space.

Fields of fact space can be defined similarly.

Both binary operations in a field are binary bijections.

Example 22.15: (1) $\langle \mathbf{Q}, +, \bullet \rangle$, $\langle \mathbf{R}, +, \bullet \rangle$, $\langle \mathbf{C}, +, \bullet \rangle$ are all fields of term space.

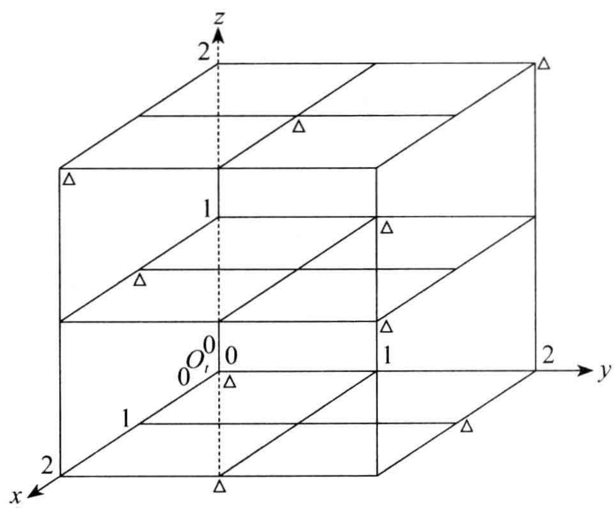


Fig. 22.7 Mutually inverse diagram for $+_3$

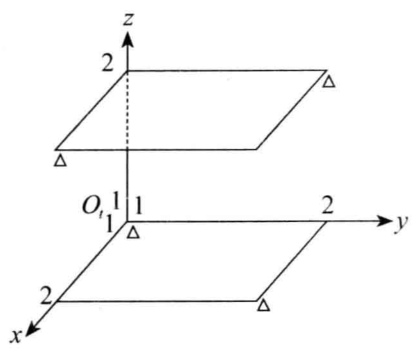


Fig. 22.8 Mutually inverse diagram for $\times_3$

(2) $\langle \mathbf{N}_k, +_k, \times_k \rangle$ is a field of term space, if and only if k is a prime number. Of the field of term space $\langle \mathbf{N}_3, +_3, \times_3 \rangle$, the mutually inverse diagram of $+_3$ is shown in Fig. 22.7, that of $\times_3$ is shown in Fig. 22.8. both are binary bijections.

(3) $\langle \{[0]_3, [1]_3, [2]_3\}, +_3, \times_3 \rangle$ is a field of fact space.

22.4.3 Boolean sum-Boolean product algebra

Boolean sum-Boolean product algebra is a binary bijective auxiliary algebra of fact space.

Definition 22.18: $\langle \rho(S), \sim, -1, \oplus, \otimes, \varnothing, S \rangle$ is Boolean sum-Boolean product algebra, whose definition is as follows:

(1) $\langle \rho(S), \oplus, -1, \varnothing \rangle$ is an Abelian group of fact space, where $\varnothing$ is the identity of Boolean sum.

(2) $\langle \rho(S), \otimes, -1, S \rangle$ is an Abelian group of fact space, where S is the identity of Boolean product.

(3) $\oplus$ and $\otimes$ are linked by the following complementary absorption laws:

$P\otimes\{P\oplus Q\}=^{-1}\sim Q$,

$P\oplus\{P\otimes Q\}=^{-1}\sim Q$.

(4) Any element $P\in\rho(S)$ has a unique complement $\sim P$.

$\langle \rho(S), \sim, -1, \oplus, \otimes, \varnothing, S \rangle$	satisfies the following laws:
$P\otimes\{P\oplus Q\}=^{-1}\sim Q$	Complement absorption law
$P\oplus\{P\otimes Q\}=^{-1}\sim Q$	Complement absorption law
$\sim\sim P=^{-1}P$	Double negation law
$\sim\varnothing=^{-1}S$	Zero-one law
$\sim S=^{-1}\varnothing$	Zero-one law
$P\oplus\sim P=^{-1}S$	Law of excluded middle and non-contradiction
$P\otimes\sim P=^{-1}\varnothing$	Law of anti-excluded middle and non-contradiction
$\varnothing\otimes P=^{-1}\sim P$	Reversion law
$S\oplus P=^{-1}\sim P$	Reversion law
$\sim\{P\oplus Q\}=^{-1}P\otimes Q$	Complementary De Morgan's law
$\sim\{P\otimes Q\}=^{-1}P\oplus Q$	Complementary De Morgan's law
$P\oplus P=^{-1}\varnothing$	Law of inverse element
$P\otimes P=^{-1}S$	Law of inverse element
$\varnothing\oplus P=^{-1}P$	Law of identity
$S\otimes P=^{-1}P$	Law of identity
$P\oplus Q=^{-1}Q\oplus P$	Commutative law
$P\otimes Q=^{-1}Q\otimes P$	Commutative law
$\{P\oplus Q\}\oplus R=^{-1}P\oplus\{Q\oplus R\}$	Associative law

$\{P \otimes Q\} \otimes R =^{-1} P \otimes \{Q \otimes R\}$ Associative law

$\sim P \oplus \sim Q =^{-1} P \oplus Q$

$\sim P \otimes \sim Q =^{-1} P \otimes Q$

$P \otimes Q =^{-1} \sim P \oplus Q$

$P \otimes Q =^{-1} P \oplus \sim Q$

$P \oplus Q =^{-1} \sim P \otimes Q$

$P \oplus Q =^{-1} P \otimes \sim Q$

$\{P \oplus Q =^{-1} R\} \subseteq^{-1} \{P \oplus R =^{-1} Q\} \cap \{Q \oplus R =^{-1} P\}$ Swap law

$\{P \otimes Q =^{-1} R\} \subseteq^{-1} \{P \otimes R =^{-1} Q\} \cap \{Q \otimes R =^{-1} P\}$ Swap law

Now, we use mutually inverse diagram to prove the complementary absorption law $P \otimes \{P \oplus Q\} =^{-1} \sim Q$. In Fig. 22.9, the compartments with oblique lines denote $P \oplus Q$, the compartments with vertical lines denote $P \otimes \{P \oplus Q\}$, which is just $\sim Q$.

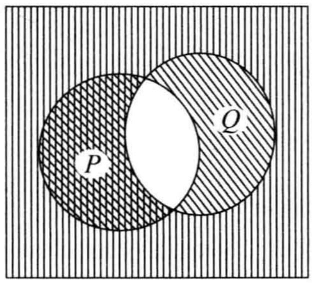


Fig. 22.9 The proof of $P \otimes \{P \oplus Q\} =^{-1} \sim Q$

The law of excluded middle and non-contradiction $P \oplus \sim P =^{-1}$S denotes that, of P and $\sim P$, exactly one holds (law of excluded middle), one does not hold (law of non-contradiction). The law of anti-excluded middle and non-contradiction $P \otimes \sim P =^{-1} \varnothing$ denotes that, of P and $\sim P$, it is not possible for them to hold simultaneously (law of non-contradiction), and it is not possible for them not to hold simultaneously (law of excluded middle).

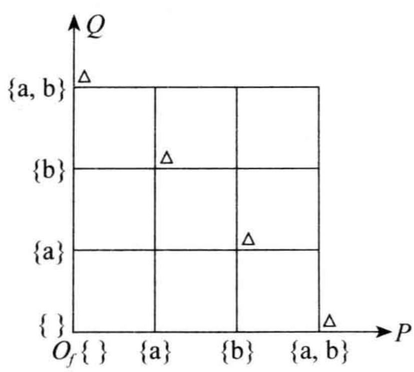


Fig. 22.10 Mutually inverse diagram for $Q =^{-1} \sim P$

The mutually inverse diagram of $\oplus$ in $\langle\rho(S), \sim, -1, \oplus, \otimes, \varnothing, S\rangle$ has been shown in Fig. 22.4, $\otimes$ can be studied dually. The mutually inverse diagram of -1 has been shown in Fig. 22.5. Now, we show the mutually inverse diagram of $\sim$ in Fig. 22.10, which is a transaxis straight line denoting the derivative $\frac{dQ}{dP}=-1$, revealing that the complement of P is $\sim P$ $(Q=^{-1}\sim P)$.

22.5 Idempotent auxiliary algebras

22.5.1 Idempotent auxiliary algebras of one binary operation

Definition 22.19: The algebra $\langle I, *\rangle$, where $I\subseteq^{-1} I_0$ and $*$ is a binary operation, satisfying idempotency and commutativity, is called an idempotent auxiliary algebra of term space.

Similarly, we can define idempotent auxiliary algebras of fact space.

Definition 22.20: Suppose $\langle I, *\rangle$ is an idempotent auxiliary algebra of term space, if $*$ satisfies associativity, then $\langle I, *\rangle$ is an associative idempotent auxiliary algebra of term space.

Similarly, we can define associative idempotent auxiliary algebras of fact space.

Example 22.16: (1) Both $\langle\{1, 2, 3, 6\}, \mathrm{GCD}\rangle$ and $\langle\{1, 2, 3, 6\}$, LCM (least common multiplier)$\rangle$ are associative idempotent auxiliary algebras of term space.

(2) Both $\langle \mathbf{N}_k, \min\rangle$ and $\langle \mathbf{N}_k, \max\rangle$ are associative idempotent auxiliary algebras of term space.

(3) Both $\langle\{F, T\}, \wedge\rangle$ and $\langle\{F, T\}, \vee\rangle$ are associative idempotent auxiliary algebras of fact space.

(4) Both $\langle\rho(S), \cap\rangle$ and $\langle\rho(S), \cup\rangle$ are associative idempotent auxiliary algebras of fact space. The mutually inverse diagram of $\cap$ is shown in Fig. 22.11. In Fig. 22.11, the transaxis straight line A denotes the partial derivative $\frac{\partial R}{\partial P}\big|_{Q=^{-1}\varnothing}=0$, reveals that when $Q=^{-1}\varnothing$, whatever P changes, R remains $\varnothing$ constantly; that $P\cap\varnothing=^{-1}\varnothing$, that $Q=^{-1}\varnothing$ is the right zero element with regard to $\cap$. The transaxis straight line B denotes $\frac{\partial R}{\partial Q}\big|_{P=^{-1}\varnothing}=0$, reveals that $P=^{-1}\varnothing$ is the left zero element with regard to $\cap$. The transaxis straight line C denotes $\frac{\partial R}{\partial Q}\big|_{P=^{-1}S}=1$, reveals that $S\cap Q=^{-1}Q$, that $P=^{-1}S$ is the left identity with regard to $\cap$. The transaxis straight line D denotes $\frac{\partial R}{\partial P}\big|_{Q=^{-1}S}=1$, reveals that $Q=^{-1}S$ is the right identity with regard to $\cap$. The transaxis straight line F denotes $\frac{\partial Q}{\partial P}\big|_{R=^{-1}\varnothing}=-1$, reveals that $P\cap\sim P=^{-1}\varnothing$ (law of non-contradiction)

where $\sim P=^{-1}Q$. The transaxis straight line E denotes the directional derivative $D_u\cap(P, Q)=\frac{\partial R}{\partial P}\cos\theta+\frac{\partial R}{\partial Q}\sin\theta=\sqrt{2}$, reveals idempotent law $P\cap P=^{-1}P$. If we view the transaxis straight lines A, C, and E as vectors $\vec{A}$, $\vec{C}$, and $\vec{E}$ respectively, then $\vec{E}=\vec{A}+\vec{C}$. Likewise, $\vec{E}=\vec{B}+\vec{D}$. $\cup$ can be studied dually.

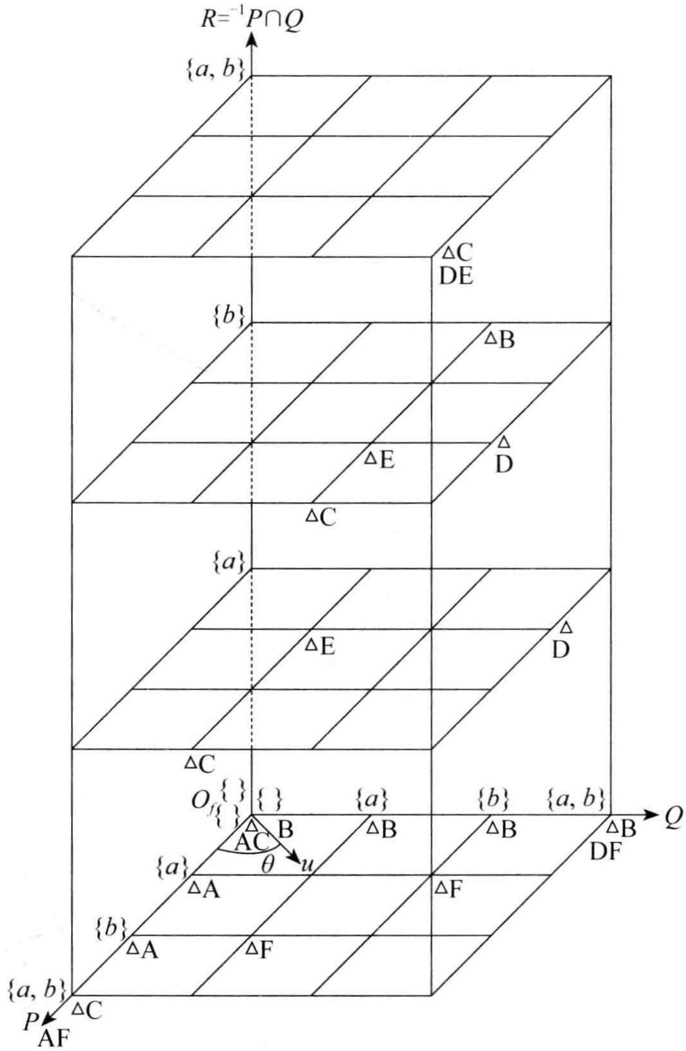


Fig. 22.11 Mutually inverse diagram for $\cap$

22.5.2 Lattices

Lattices are idempotent auxiliary algebras of two binary operations.

Definition 22.21: Suppose ⟨L, $*$⟩ and ⟨L, $+$⟩ are two associative idempotent auxiliary algebras of term space, and $*$ and $+$ satisfy the absorption laws:

$x*(x+y)=x$

$x+(x*y)=x$

then ⟨L, ∗, +⟩ is a lattice of term space.

Similarly, we can define lattices of fact space.

22.6 Complementary idempotent auxiliary algebras

22.6.1 Complementary idempotent auxiliary algebras of one binary operation

This section studies complementary idempotent auxiliary algebras of one binary operation of fact space. Complementary idempotent auxiliary algebras of one binary operation of term space will be studied in Chapter 38.

NAND ↑ is defined as $\sim\{P\cap Q\}$, NOR ↓ is defined as $\sim\{P\cup Q\}$. ↑ and ↓ satisfy neither binary bijection nor idempotency.

Definition 22.22: Suppose we have the algebra ⟨C, ⊙⟩, where $C\subseteq^{-1}I_1$, and ⊙ is a binary operation. If ⊙ satisfies complementary idempotent law

$P\odot P=^{-1}\sim P$

and commutative law

$P\odot Q=^{-1}Q\odot P$

then ⟨C, ⊙⟩ is called a complementary idempotent auxiliary algebra of fact space.

Example 22.17: ⟨ρ(S), ~, ~−1, ↑, ∅, S⟩ is a complementary idempotent auxiliary algebra of fact space, satisfies

$P\uparrow P=^{-1}\sim P$ Complementary idempotent law

$P\uparrow Q=^{-1}Q\uparrow P$ Commutative law

$P\uparrow S=^{-1}\sim P$ S is the complementary identity of ↑

$P\uparrow \varnothing=^{-1}\sim\varnothing$ ∅ is the complementary zero element of ↑

$P\uparrow \sim P=^{-1}S$ P and ~P are mutually complements and complementary inverse elements

The mutually inverse diagram of ↑ is shown in Fig. 22.12. In Fig. 22.12, the transaxis straight line A denotes $\frac{\partial R}{\partial P}\Big|_{Q=^{-1}\varnothing}=0$, reveals that when $Q=^{-1}\varnothing$, whatever P changes R remains S constantly, that $P\uparrow\varnothing=^{-1}S=^{-1}\sim\varnothing$, that $Q=^{-1}\varnothing$ is the right complementary zero element with regard to ↑. The transaxis straight line B denotes $\frac{\partial R}{\partial Q}\Big|_{P=^{-1}\varnothing}=0$, reveals that $P=^{-1}\varnothing$ is the left complementary zero element with regard to ↑. The transaxis straight line C denotes $\frac{\partial R}{\partial Q}\Big|_{P=^{-1}S}=-1$, reveals that $P=^{-1}S$ is the left complementary identity with regard to ↑. The transaxis straight line D denotes $\frac{\partial R}{\partial P}\Big|_{Q=^{-1}S}=-1$, reveals that $Q=^{-1}S$ is the right

complementary identity with regard to ↑. The transaxis straight line F denotes $\frac{\partial Q}{\partial P}\Big|_{R=^{-1}S}=-1$, reveals that $P\uparrow\sim P=^{-1}S(\sim P=^{-1}Q)$, that $\sim P$ is the complement and complementary inverse element of P. The transaxis straight line E denotes the directional derivative $D_u\uparrow(P, Q)=\frac{\partial R}{\partial P}\cos\theta+\frac{\partial R}{\partial Q}\sin\theta=\sqrt{2}$, reveals complementary idempotent law $P\cap P=^{-1}\sim P$. If we view the

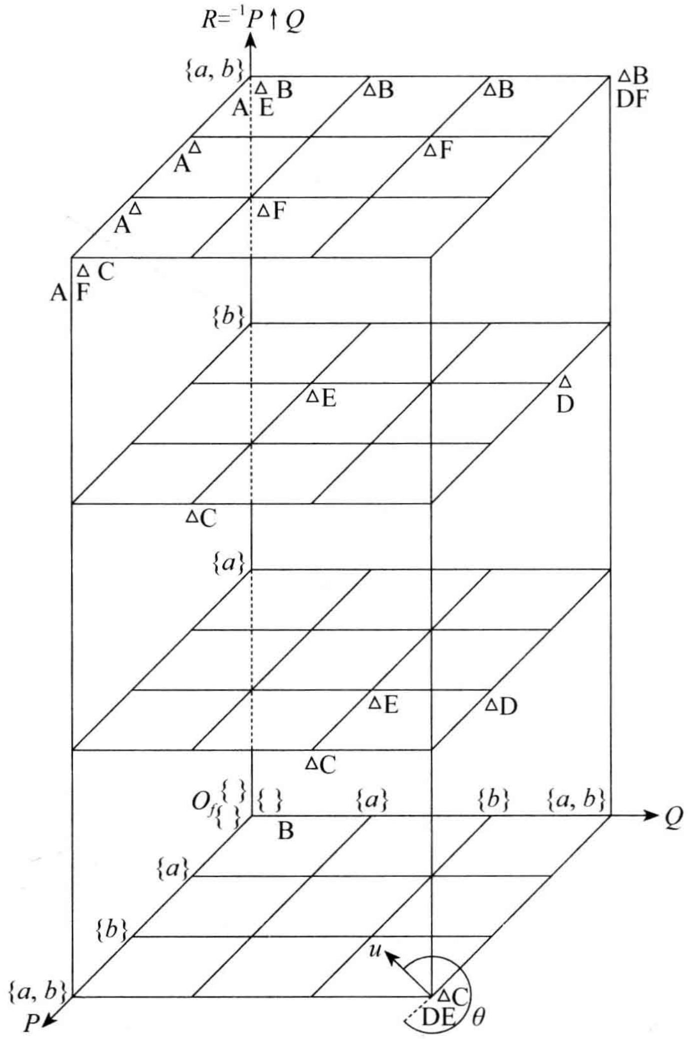


Fig. 22.12 Mutually inverse diagram for ↑

transaxis straight lines A, C, and E as vectors $\vec{A}$, $\vec{C}$, and $\vec{E}$ respectively, then $\vec{E}=\vec{A}+\vec{C}$. Likewise, $\vec{E}=\vec{B}+\vec{D}$. The mutually inverse diagram of $\sim$−1 is shown in Fig. 22.13, the transaxis straight line of which denotes $\frac{dQ}{dP}=-1$. Reveals that the complementary inverse element $P^{\sim-1}$ of P is $\sim P$.

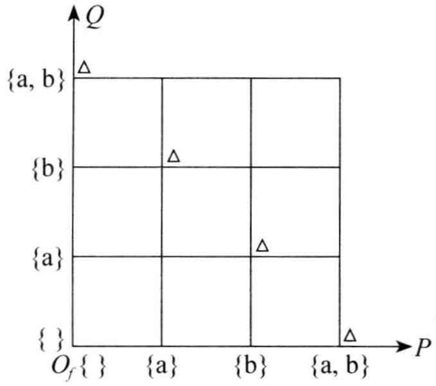


Fig.22.13 Mutually inverse diagram for $Q=^{-1}P^{\sim-1}$

Example 22.18: ⟨ρ(S), ∼, ∼−1, ↓, ∅, S⟩ is a complementary idempotent auxiliary algebra of fact space, satisfies

$P\downarrow P=^{-1}\sim P$ Complementary idempotent law

$P\downarrow Q=^{-1}Q\downarrow P$ Commutative law

$P\downarrow \varnothing=^{-1}\sim P$ ∅ is the complementary identity of ↓

$P\downarrow \mathrm{S}=^{-1}\sim \mathrm{S}$ S is the complementary zero element of ↓

$P\downarrow \sim P=^{-1}\varnothing$ P and ∼P are mutually complements and complementary inverse elements

22.6.2 NAND-NOR algebra

NAND-NOR algebra is a complementary idempotent auxiliary algebra of two binary operations of fact space.

Definition 22.23: If ⟨ρ(S), ∼, ∼−1, ↑, ↓, ∅, S⟩ satisfies

(1) ⟨ρ(S), ∼, ∼−1, ↑, ∅, S⟩ is a complementary idempotent auxiliary algebra.

(2) ⟨ρ(S), ∼, ∼−1, ↓, ∅, S⟩ is a complementary idempotent auxiliary algebra.

(3) ↑ and ↓ are linked by the following complementary distributive laws:

$$P\uparrow\{Q\downarrow R\}=^{-1}\sim\{\{P\uparrow\sim Q\}\downarrow\{P\uparrow\sim R\}\}$$

$$P\downarrow\{Q\uparrow R\}=^{-1}\sim\{\{P\downarrow\sim Q\}\uparrow\{P\downarrow\sim R\}\}$$

then ⟨ρ(S), ∼, ∼−1, ↑, ↓, ∅, S⟩ is a NAND-NOR algebra.

NAND-NOR algebra also satisfies De Morgan's laws:

$\sim\{P\uparrow Q\}=^{-1}\sim P\downarrow\sim Q$,

$\sim\{P\downarrow Q\}=^{-1}\sim P\uparrow\sim Q$.

Example 22.19: Prove complementary distributive law $P\downarrow\{Q\uparrow R\}=^{-1}\sim\{\{P\downarrow\sim Q\}\uparrow\{P\downarrow\sim R\}\}$.

Proof: The mutually inverse diagram of $Q\uparrow R$ is shown as the compartments with horizontal lines in Fig. 22.14. The mutually inverse diagram of $P\downarrow\{Q\uparrow R\}$ is shown as the compartments with vertical lines in Fig. 22.14, which is $\sim P\cap Q\cap R$. Therefore, we have

$P \downarrow \{Q \uparrow R\}$
$=^{-1} \sim P \cap Q \cap R$
$=^{-1} \sim\sim \{\sim P \cap Q \cap R\}$
$=^{-1} \sim \{P \cup \sim Q \cup \sim R\}$
$=^{-1} \sim \{\{P \cup \sim Q\} \cup \{P \cup \sim R\}\}$
$=^{-1} \sim \{P \cup \sim Q\} \cap \sim \{P \cup \sim R\}$
$=^{-1} \{P \downarrow \sim Q\} \cap \{P \downarrow \sim R\}$
$=^{-1} \sim\sim \{\{P \downarrow \sim Q\} \cap \{P \downarrow \sim R\}\}$
$=^{-1} \sim \{\{P \downarrow \sim Q\} \uparrow \{P \downarrow \sim R\}\}$
Q.E.D.

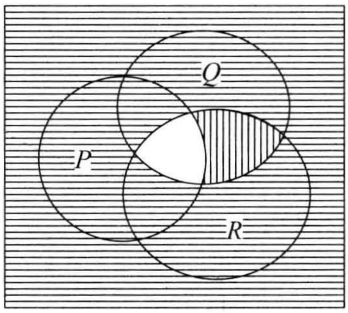


Fig. 22.14 Mutually inverse diagram for $P \downarrow \{Q \uparrow R\} =^{-1} \sim \{\{P \downarrow \sim Q\} \uparrow \{P \downarrow \sim R\}\}$

The complementary distributive law $P \uparrow \{Q \downarrow R\} =^{-1} \sim \{\{P \uparrow \sim Q\} \downarrow \{P \uparrow \sim R\}\}$ can be proved similarly.

By the way, nmin-nmax algebra and NGCD-NLCM algebra are complementary idempotent auxiliary algebras of two binary operations of term space.

22.7 Isomorphism between algebras

The sameness of structures between two algebras is called isomorphism, which can be described by unary bijections connecting the corresponding operations and constants of the two algebras. Here, we only discuss the two algebras in the form of A=⟨S, ∗, k⟩ and A'=⟨S', ∗', k'⟩, where ∗ and ∗' are binary operations, k and k' are constants.

Definition 22.24: The algebras A=⟨S, ∗, k⟩ and A'=⟨S', ∗', k'⟩ are isomorphic, if there exists a unary bijection h, such that

(1) h:S→S'

(2) h(a∗b)=h(a)∗'h(b)

(3) h(k)=k'.

Here, a, b are arbitrary elements in S. Condition (2) is called that the operations preserve under h.

If the mutually inverse diagrams of $*$ and $*'$ are the same, and a unary bijection is established between the coordinates of the mutually inverse diagram of $*$ and the corresponding coordinates of the mutually inverse diagram of $*'$, then the isomorphism between A=<S, $*$, k> and A'=<S', $*'$, k'> can be established.

Example 22.20: The mutually inverse diagram of the operation GCD of the algebra <{1, 2, 3, 6}, GCD> has been shown in Fig. 10.5. The mutually inverse diagram of the operation ∩ of the algebra <ρ({a, b}), ∩> has been shown in Fig. 22.11. The two mutually inverse diagrams are identical. The establishment of the unary bijection between their corresponding coordinates is as follows:

h(1)={}

h(2)={a}

h(3)={b}

h(6)={a, b}

then isomorphism is established between <{1, 2, 3, 6}, GCD> and <ρ({a, b}), ∩>. Now, we use 2 and 3 to verify. We do operation first, then do mapping: h(GCD(2, 3))=h(1)={}. We do mapping first, then do operation: $h(2)\cap h(3)=^{-1}\{a\}\cap\{b\}=^{-1}\{\}$. The two results are identical; i.e., the operations preserve under h.

Example 22.21: The mutually inverse diagram of ⊕ of the algebra <ρ({a}), ⊕> has been shown in Fig. 10.12. The mutually inverse diagram of ⊗ of the algebra <ρ({a}), ⊗> has been shown in Fig. 10.13. The two mutually inverse diagrams are not identical. But we can rearrange the order of the coordinates of the mutually inverse diagram of ⊗, such that the two mutually inverse diagrams are identical. After the rearrangement, Fig. 10.13 becomes Fig. 22.15, which is identical with Fig. 10.12. The establishment of the unary bijection between their coordinates is as follows:

h({})={a}

h({a})={}

then isomorphism can be established between <ρ({a}), ⊕> and <ρ({a}), ⊗>.

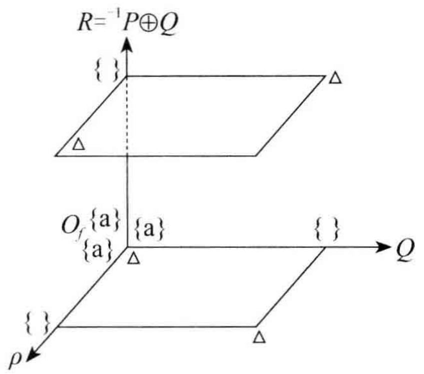


Fig. 22.15 Mutually inverse diagram of ⊗ after rearrangement

22.8 Summary on transaxis straight lines-derivatives-algebraic properties

Transaxis straight lines belong to geometry, derivatives belong to analysis, algebraic properties belong to algebra. While in this chapter, the three are integrated organically.

The derivatives of zero elements and complementary zero elements are always 0, they are transaxis straight lines spanning over one axis. The derivatives of identities are always 1, they are transaxis straight lines spanning over two axes. The derivatives of complementary identities are always −1, they are transaxis straight lines spanning over two axes. The derivatives of idempotent laws and complementary idempotent laws are always $\sqrt{2}$, they are transaxis straight lines spanning over three axes.

22.9 Comparisons among various algebras of two binary operations

Boolean algebras have complements but have no inverse elements. Rings and fields have inverse elements but have no complements. Boolean sum-Boolean product algebra has both complements and inverse elements. NAND-NOR algebra has complements and complementary inverse elements but has no inverse elements.

Rings and fields have additive identities, multiplicative identities and zero elements, the additive identities are just multiplicative zero elements. In Boolean algebra, $\varnothing$ is the identity of $\cup$ and the zero element of $\cap$, S is the zero element of $\cup$ and the identity of $\cap$. Boolean sum-Boolean product algebra has identity of Boolean sum and identity of Boolean product, but has no zero element. NAND-NOR algebra has neither identity nor zero element, but has complementary identity and complementary zero element; S is the complementary identity of $\uparrow$ and the complementary zero element of $\downarrow$, $\varnothing$ is the complementary identity of $\downarrow$ and the complementary zero element of $\uparrow$.

In rings and fields, + and • are linked by the distributive law. In lattices, $\cap$ and $\cup$ are linked by the absorption law. In Boolean sum-Boolean product algebra, $\oplus$ and $\otimes$ are linked by the complementary absorption law. In NAND-NOR algebra, $\uparrow$ and $\downarrow$ are linked by the complementary distributive law.

+ and • make Abelian groups of term space respectively, make a field of term space jointly. $\cap$ and $\cup$ make associative idempotent auxiliary algebras of fact space respectively, makes lattice and Boolean algebra jointly. $\oplus$ and $\otimes$ make Abelian groups of fact space respectively, makes Boolean sum-Boolean product algebra jointly. $\uparrow$ and $\downarrow$ make complementary idempotent auxiliary algebras of fact space respectively, makes NAND-NOR algebra jointly.

22.10 The laws various auxiliary algebras of one binary operation satisfy

Groups must satisfy associative law, can satisfy commutative law. Idempotent auxiliary algebras must satisfy commutative law, can satisfy associative law. Complementary idempotent auxiliary algebra must satisfy commutative law, must not satisfy associative law.

22.11 Comparison between auxiliary algebras and classical abstract algebra

Compared with classical abstract algebra, auxiliary algebras have the following advantages.

(1) Boolean sum-Boolean product algebra and NAND-NOR algebra are proposed by the author.

(2) Various auxiliary algebras are described by mutually inverse diagrams, reflecting intuitively binary bijection, idempotency, complementary idempotency, commutativity, in which binary bijection and complementary idempotency are discovered by the author.

Chapter 23 Main-auxiliary algebras

The difference between main-auxiliary algebras and auxiliary algebras is that in the main-auxiliary algebras the partially ordered main plays an important role.

23.1 Lattices

23.1.1 Lattices as partially ordered sets

It can be proved that if there exists the least upper bound (lub) for the subset of a partially ordered set, then it is unique; and if there exists the greatest lower bound (glb), then it is also unique. Now, we introduce lattices based on this.

Definition 23.1: Suppose $\langle L, \leqslant\rangle$ is a partially ordered set, where $L \subseteq^{-1} I_0$. If there exist the greatest lower bound and the least upper bound for any couple of elements x, y in L, then $\langle L, \leqslant\rangle$ is a lattice of term space.

Usually, $x*y$ is used to denote the greatest lower bound of $\{x, y\}$, and $x+y$ is used to denote the least upper bound of $\{x, y\}$. That is,

$x*y=\text{glb}\{x, y\}$

$x+y=\text{lub}\{x, y\}$.

They are called meet and join of x and y respectively. Because the greatest lower bound and the least upper bound belong to L, and are unique, both meet $*$ and join $+$ are the binary operations on L.

Definition 23.2: Suppose $\langle L, \leqslant^{-1}\rangle$ is a partially ordered set, where $L \subseteq^{-1} I_1$. If there exist the greatest lower bound and the least upper bound for any couple of elements P, Q in L, then $\langle L, \leqslant^{-1}\rangle$ is a lattice of fact space.

Usually, $P \wedge Q$ is used to denote the greatest lower bound of $\{P, Q\}$, and $P \vee Q$ is used to denote the least upper bound of $\{P, Q\}$. That is,

$P \wedge Q =^{-1} \text{glb}\{P, Q\}$

$P \vee Q =^{-1} \text{lub}\{P, Q\}$.

They are called meet and join of P and Q respectively.

Example 23.1:

(1) Suppose D is the dividing evenly relation on the set of positive integers $\mathbf{Z}_+$, then $\langle \mathbf{Z}_+, D\rangle$ is a lattice of term space, because for any $x, y \in \mathbf{Z}_+$:

$x*y=\text{glb}\{x, y\}=\text{GCD}\{x, y\}$

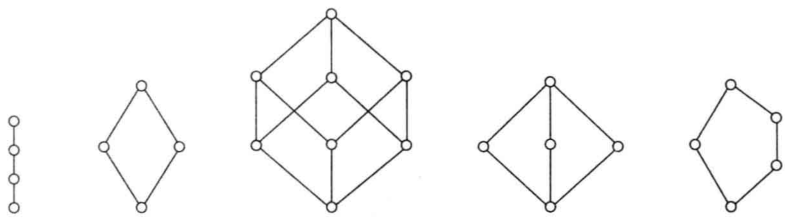
$x+y=\text{lub}\{x, y\}=\text{LCM}\{x, y\}$

(2) Suppose n is a positive integer, S_n is the set of all of the factors of n. For example, $S_6=\{1, 2, 3, 6\}$, $S_8=\{1, 2, 4, 8\}$. D is the dividing evenly relation. Then $\langle S_n, D\rangle$ is a lattice of term space. The Hasse diagrams of $\langle S_8, D\rangle$, $\langle S_6, D\rangle$, $\langle S_{30}, D\rangle$ are shown in Fig. 23.1 (a), (b), and (c) respectively.

(3) Suppose S is any set, ρ(S) is its power set, then the partially ordered set $\langle \rho(S), \subseteq^{-1}\rangle$ is a lattice of fact space. Because for any subsets P and Q of S, $P\cup Q$ is the least upper bound of P and Q, $P\cap Q$ is the greatest lower bound of P and Q. When S has two or three elements, the corresponding lattices are also shown in Fig. 23.1 (b) and (c) respectively.

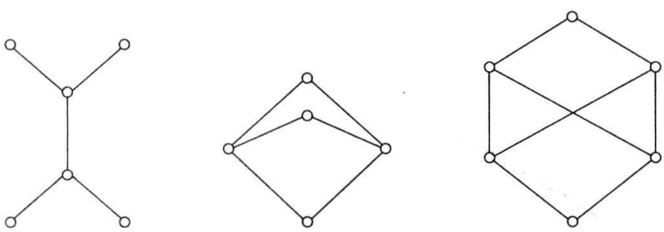
(4) The Hasse diagrams shown in Fig. 23.1 (d) and (e) are also lattices.

(5) The Hasse diagrams shown in Fig. 23.2 (a), (b), and (c) are not lattices.



(a) (b) (c) (d) (e)

Fig. 23. 1 Hasse diagrams of lattices



(a) (b) (c)

Fig. 23.2 Hasse fiagrams of non-lattices

23.1.2 Lattices made from idempotent auxiliary algebras

Definition 23.3: Suppose $\langle L, *\rangle$ and $\langle L, +\rangle$ are two associative idempotent auxiliary algebras of term space, and $*$ and $+$ satisfy the absorption laws, then $\langle L, *, +\rangle$ is a lattice of term space.

Similarly, we can define lattice of fact space.

It can be proved that Definitions 23.1 and 23.3 are equivalent. The lattice made from idempotent auxiliary algebras belongs to auxiliary algebras. The lattice made from partially ordered sets belongs to main-auxiliary algebras.

23.1.3 Special lattices

Definition 23.4: Suppose ⟨L, $*$, +⟩ is a lattice of term space, if for any x, y, $z \in$L, we have

$x*(y+z)=(x*y)+(x*z)$

$x+(y*z)=(x+y)*(x+z)$

then we say that ⟨L, $*$, +⟩ is a distributive lattice of term space.

Similarly, we can define distributive lattice of fact space.

Example 23.2: None of the two lattices shown in Fig. 23.3 is a distributive lattice.

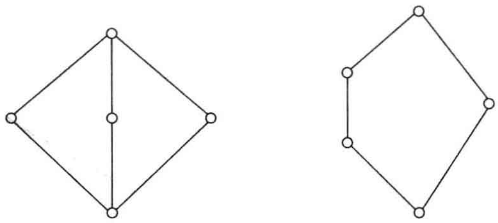


Fig. 23.3 Hasse diagrams for non-distributive lattices

Definition 23.5: If there exists an element a in the lattice of term space ⟨L, $\leqslant$⟩, such that for any element x, we have a$\leqslant x$($x\leqslant$a), then a is called the total lower bound (total upper bound) of the lattice ⟨L, $\leqslant$⟩.

Similarly, we can define the total lower bound (total upper bound) of a lattice of fact space.

It can be proved that the total lower bound (total upper bound) is unique.

Example 23.3: In the lattice of fact space ⟨ρ(S), $\subseteq^{-1}$⟩, S is the total upper bound, ∅ is the total lower bound.

Definition 23.6: If there exist the total upper bound and total lower bound for a lattice, then they are called the bounds of the lattice, and are denoted by 0(**0**) and 1(**1**) respectively. A lattice having 0(**0**) and 1(**1**) is called a bounded lattice.

Example 23.4: Any finite lattice is necessarily a bounded lattice.

Definition 23.7: Suppose ⟨L, $*$, +, 0, 1⟩ is a bounded lattice of term space. If for some element a$\in$L, there exists element b$\in$L, such that

a$*$b=0 a+b=1

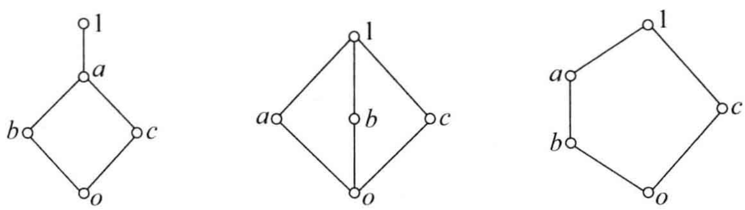
then we say that b is the complement of a, denoted by a'.

If b is the complement of a, then a is also the complement of b. Generally speaking, a$\in$L may not have a complement. If it has, its complement may not be unique.

Example 23.5: In Fig. 23.4 (a), none of a, b, and c has complement. In Fig. 23.4 (b), a, b, and c are mutually complements, their complements are not unique. In Fig. 23.4 (c), the complements of c are a and b, the complement of a is c, the complement of b is c.

Definition 23.8: In a bounded lattice, if every element has at least one complement, then the lattice is called a complemented lattice.

Example 23.6: Fig. 23.4 (b) and (c) are complemented lattices, while Fig. 23.4 (a) is not a complemented lattice.



(a) (b) (c)

Fig. 23.4 Complements and complemented lattices

23.2 Boolean algebras

Definition 23.9: If a lattice is both complemented and distributive, then it is a Boolean algebra.

It can be proved that in a Boolean algebra, the complement of every element is unique, therefore we can define a unary operation—complement operation in a Boolean algebra. Thus, a Boolean algebra is an algebraic system that has two binary operations and one unary operation. A Boolean algebra of term space is denoted as ⟨B, $*$, $+$, ’, 0, 1⟩. A Boolean algebra of fact space is denoted as ⟨B, $\wedge$, $\vee$, $\neg$, **0**, **1**⟩.

A Boolean algebra of term space satisfies the following properties:

(1) ⟨B, $*$, $+$⟩ is a lattice, satisfies

$x*x=x$

$x+x=x$

$x*y=y*x$

$x+y=y+x$

$(x*y)*z=x*(y*z)$

$(x+y)+z=x+(y+z)$

$x*(x+y)=x$

$x+(x*y)=x$

(2) ⟨B, $*$, $+$⟩ is a distributive lattice, satisfies

$x*(y+z)=(x*y)+(x*z)$

$x+(y*z)=(x+y)*(x+z)$

$(x*y)+(y*z)+(z*x)=(x+y)*(y+z)*(z+x)$

$(x*y=x*z)\wedge(x+y=x+z)\leqslant^{-1}y=z$

(3) $\langle B, *, +, 0, 1\rangle$ is a bounded lattice, satisfies

$0 \leqslant x \leqslant 1$

$x*0=0$

$x+1=1$

$x*1=x$

$x+0=x$

(4) $\langle B, *, +, ', 0, 1\rangle$ is a complemented lattice, satisfies

$x*x'=0$

$x+x'=1$

$0'=1$

$1'=0$

(5) $\langle B, *, +, ', 0, 1\rangle$ is a complemented and distributive lattice, satisfies

$(x*y)'=x'+y'$

$(x+y)'=x'*y'$

(6) There exists the partial ordering $\leqslant$ in the set B, satisfying

$x*y=\mathrm{glb}\{x, y\}$

$x+y=\mathrm{lub}\{x, y\}$

$x \leqslant y =^{-1} x*y=x =^{-1} x+y=y$

$x \leqslant y =^{-1} x*y'=0 =^{-1} x'+y=1 =^{-1} y' \leqslant x'$

A Boolean algebra of fact space satisfies similar properties.

Definition 23.10: Suppose $\langle B, *, +\rangle$ is an algebraic system of term space, $*$ and $+$ are the binary operations on B. If for any elements x, y, and z, the following four conditions are satisfied, then $\langle B, *, +\rangle$ is a Boolean algebra of term space.

(1) $x*y=y*x$ and $x+y=y+x$

(2) $x*(y+z)=(x*y)+(x*z)$ and $x+(y*z)=(x+y)*(x+z)$

(3) There exist two elements 0 and 1, such that for any element x in B

$x*1=x$ and $x+0=x$

are satisfied.

(4) For any element x in B, there exists an element x', such that

$x*x'=0$ and $x+x'=1$

A Boolean algebra of fact space can be similarly defined.

The four formulas of Definition 23.10 are basic ones, from which other formulas of a Boolean algebra of term space can be deduced. Definition 23.10 can also be used to test whether an algebraic system is a Boolean algebra of term space or not.

Example 23.7: $\langle\{F, T\}, \wedge, \vee, \neg, F, T\rangle$ is a Boolean algebra of fact space.

Example 23.8: Suppose $S=\{a, b\}$, then $\langle \rho(S), \cap, \cup, \sim, \varnothing, S\rangle$ is a Boolean algebra of fact space. The mutually inverse diagrams of $\sim$, $\cap$, and $\cup$ are shown in Figs 23.5, 23.6,

and 23.7 respectively. The Hasse diagram of the Boolean algebra is shown in Fig. 23.8.

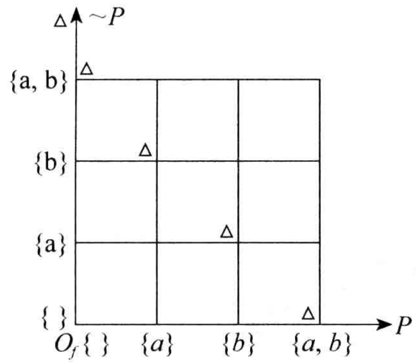


Fig. 23.5 Mutually inverse diagram for ~

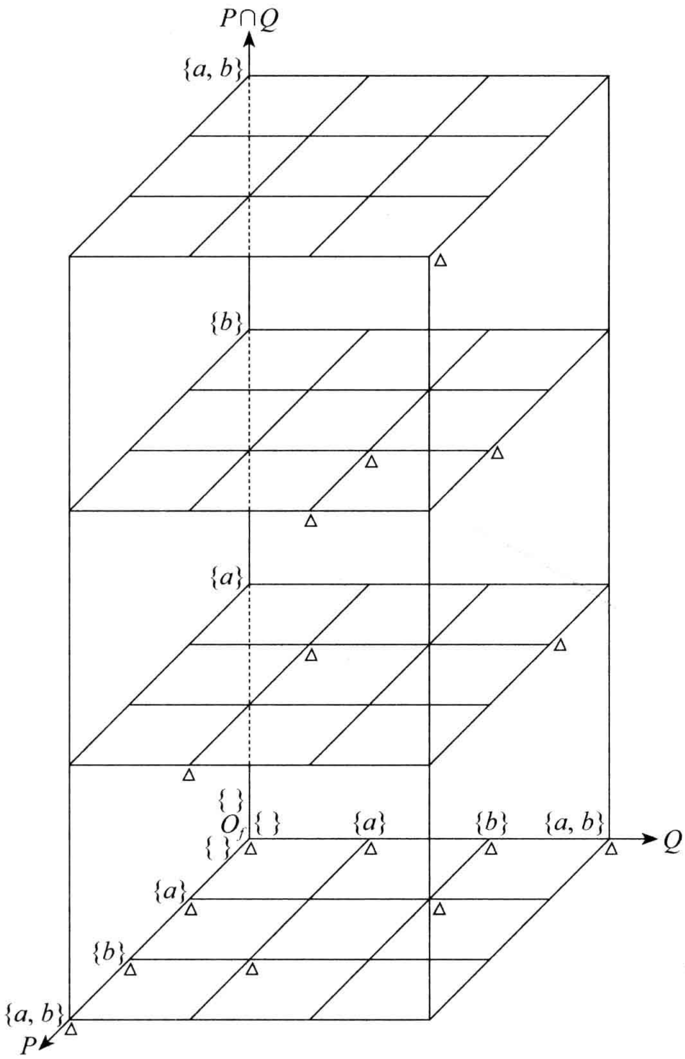


Fig. 23.6 Mutually inverse diagram for ∩

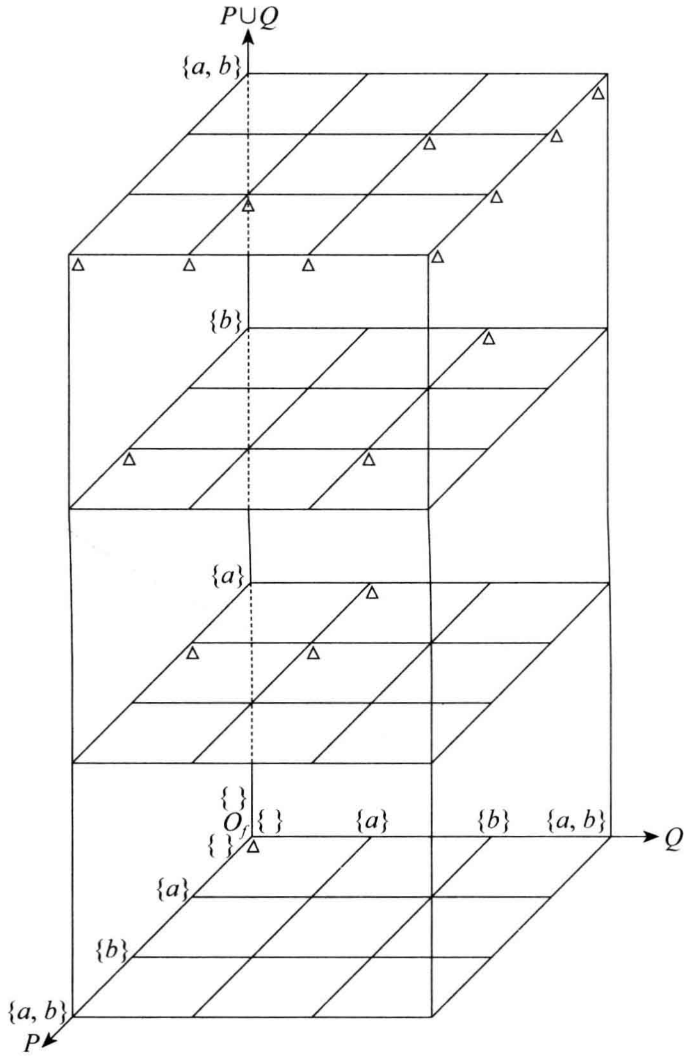


Fig. 23.7 Mutually inverse diagram for ∪

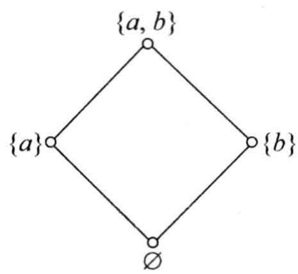


Fig. 23.8 Hasse diagram for ⟨ρ(S), ∩, ∪, ~, ∅, S⟩

Finite Boolean algebras can be constructed from atoms. Now, we define covers, atoms first, and then, we investigate how to construct Boolean algebras from atoms.

Definition 23.11: Suppose a and b are two elements in a Boolean algebra of term

space. If b≤a and b≠a; i.e., b<a, and there exists no other element c, such that b<c and c <a, then we say that element a covers element b.

Definition 23.12: Suppose ⟨B, ∗, +, ’, 0, 1⟩ is a Boolean algebra of term space, and a∈B. If a covers 0, then we say that element a is an atom of the Boolean algebra.

Every element x of a Boolean algebra of term space ⟨B, ∗, +, ’, 0, 1⟩ except 0 can be denoted as the join of atoms uniquely; i.e.,

$x=a_1+a_2+\ldots+a_k$ (a_i is an atom).

Boolean algebras of fact space have similar theory of atoms.

23.3 Main-auxiliary algebras for set theorems

Main-auxiliary algebras for set theorems are algebras of fact space.

23.3.1 Main-auxiliary algebras for quasi-set theorems

23.3.1.1 Q_1

First, let us construct the main-auxiliary algebra for quasi-set theorems with one fact proposition P: Q_1. The set of minterms of one fact proposition P is $S_1=\{\overline{P}, P\}$ ($\overline{P}$ is the same as $\sim P$). Taking every element in S_1 as an atom to construct Boolean algebra, we obtain Q_1=⟨ρ(S_1), ~, ∩, ∪, ∅, S_1⟩, shown in Fig. 23.9, where the partially ordered relation is $\subseteq^{-1}$. Each vertex in Fig. 23.9 is a fact proposition, also a unary simple-complex composition operation. Near each vertex is its mutually inverse diagram in which the shaded minterms correspond to the set of the unary simple-complex composition operation and the blank ones to its complement set. ∅ and S_1 are distinguished vertices.

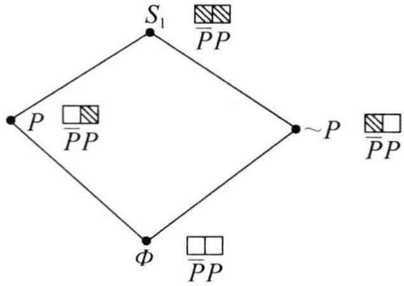


Fig. 23.9 Hasse diagram for Q_1

There are two kind of quasi-set theorems: quasi-set theorems with the empirical or mathematical connection operator being $=^{-1}$ and quasi-set theorems with the empirical or mathematical connection operator being $\subseteq^{-1}$.

A quasi-set theorem with the empirical or mathematical connection operator being $=^{-1}$

is proved in Q_1 as follows: suppose A and B are any two vertices on Q_1. If A and B are the same vertex, then $A=^{-1}B$ is a quasi-set theorem. If B is the complement of A, then $\sim A=^{-1}B$ is a quasi-set theorem. For example, $P\cup\sim P$ and S_1 are the same vertex, and therefore, $P\cup\sim P=^{-1}S_1$ is a quasi-set theorem. P is the complement of $\sim P$, and therefore, $\sim\{\sim P\}=^{-1}P$ is a quasi-set theorem.

A quasi-set theorem with the empirical or mathematical connection operator being $\subseteq^{-1}$ is proved in Q_1 as follows: suppose C and D are two vertices on Q_1; C is not $\varnothing$, D is not S_1. If D is reachable from C via an ascending path (including zero path), then $C\subseteq^{-1}D$ is a quasi-set theorem; e.g., P is reachable from P via an ascending path, and therefore, $P\subseteq^{-1}P$ is a quasi-set theorem.

23.3.1.2 Consistency and completeness theorem

Consistency: Suppose A and B are any two vertices on Q_1, C and D are two vertices on Q_1; C is not $\varnothing$, D is not S_1. If at least one of $A=^{-1}B$ and $\sim\{A=^{-1}B\}$ is not a quasi-set theorem, and at least one of $C\subseteq^{-1}D$ and $\sim\{C\subseteq^{-1}D\}$ is not a quasi-set theorem, then Q_1 is consistent.

Completeness: Suppose A and B are any two vertices on Q_1, C and D are two vertices on Q_1; C is not $\varnothing$, D is not S_1. If at least one of $A=^{-1}B$ and $\sim\{A=^{-1}B\}$ is a quasi-set theorem, and at least one of $C\subseteq^{-1}D$ and $\sim\{C\subseteq^{-1}D\}$ is a quasi-set theorem, then Q_1 is complete.

Consistency and completeness theorem for Q_1: Q_1 is both consistent and complete.

Proof: Suppose A and B are any vertices on Q_1. In the Hasse diagram of Q_1 shown in Fig. 23.9, either A and B are the same vertex, or they are not; i.e., of $A=^{-1}B$ and $\sim\{A=^{-1}B\}$, exactly one holds and one does not. Suppose C and D are any vertices on Q_1 except $\varnothing$ and S_1. In the Hasse diagram of Q_1, either D is reachable from C via an ascending path, or such a path does not exist; i.e., of $C\subseteq^{-1}D$ and $\sim\{C\subseteq^{-1}D\}$, exactly one holds and one does not. Because one of $A=^{-1}B$ and $\sim\{A=^{-1}B\}$ does not hold, and one of $C\subseteq^{-1}D$ and $\sim\{C\subseteq^{-1}D\}$ does not hold, Q_1 is consistent. Because one of $A=^{-1}B$ and $\sim\{A=^{-1}B\}$ holds, and one of $C\subseteq^{-1}D$ and $\sim\{C\subseteq^{-1}D\}$ holds, Q_1 is complete.

Q.E.D.

23.3.1.3 Q_2

Now, let us construct the main-auxiliary algebra for quasi-set theorems with two fact propositions P and Q: Q_2. The set of minterms with P and Q two fact propositions is $S_2=\{\overline{P}\overline{Q}, \overline{P}Q, P\overline{Q}, PQ\}$. Taking every element in S_2 as an atom to construct Boolean algebra, we obtain $Q_2=\langle\rho(S_2), \sim, \cap, \cup, \varnothing, S_2\rangle$, shown in Fig. 23.10, where the partially ordered relation is $\subseteq^{-1}$. Each vertex in Fig. 23.10 is a fact proposition, also a binary simple-complex composition operation. Near each vertex is its mutually inverse diagram, and the positions

of various minterms in the mutually inverse diagram are shown on the bottom left corner of Fig. 23.10. $\varnothing$ and S_2 are distinguished vertices.

The proof of quasi-set theorems on Q_2 is similar to that on Q_1. It can be proved that $P\cap\varnothing=^{-1}\varnothing$, $P\cap P=^{-1}P$, $P\cup{\sim}P=^{-1}S_2$, ${\sim}\{P\cap Q\}=^{-1}{\sim}P\cup{\sim}Q$, $P\cap Q\subseteq^{-1}P$, etc. are quasi-set theorems. Similar to Q_1, it can be proved Q_2 is both consistent and complete.

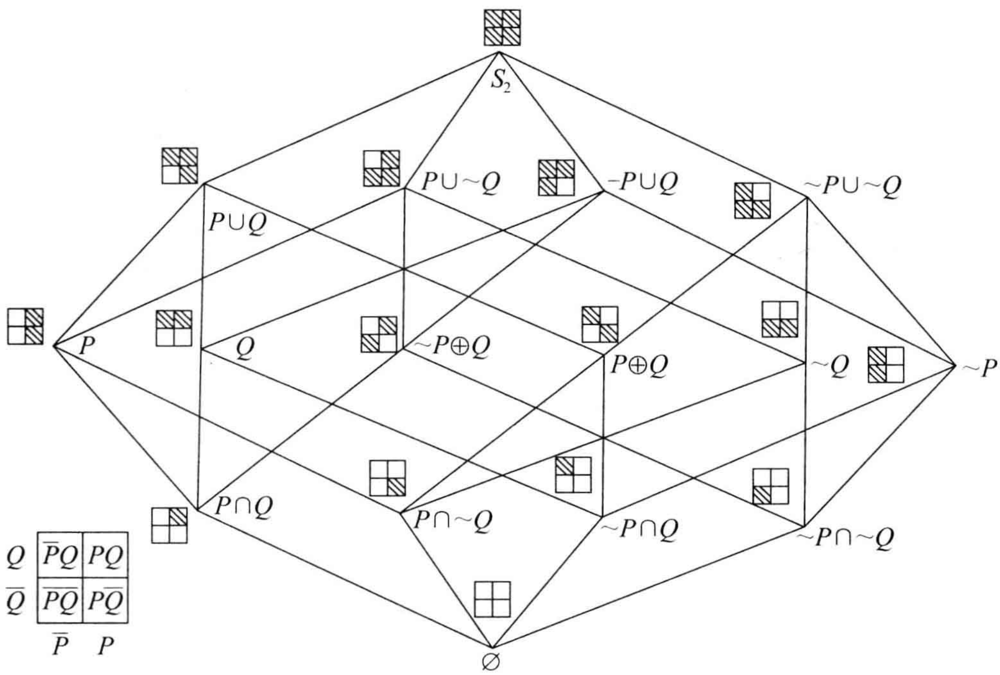


Fig. 23.10 Hasse diagram for Q_2

23.3.2 Main-auxiliary algebras for single set theorems and logical squares, logical rectangles, logical pies

23.3.2.1 US_2

The set of minterms with P and Q two fact propositions is $S_2=\{\overline{P}\overline{Q}, \overline{P}Q, P\overline{Q}, PQ\}$. Taking every element in S_2 as an atom to construct Boolean algebra, we obtain the unicellular main-auxiliary algebra for single set theorems with P and Q two fact propositions $US_2=\langle\rho(S_2), \sim, \cap, \cup, \varnothing, S_2\rangle$, shown in Fig. 23.11.

Unlike Fig. 23.10 where each vertex is a fact proposition, each vertex in Fig. 23.11 is a unicellular second-order single empirical or mathematical connection proposition. Both Q_2 and US_2 are obtained by taking every element in S_2 as an atom to construct Boolean algebra, but they are obtained differently. Take the vertex $P\oplus Q$ in Q_2 and the vertex ${\sim}P=^{-1}Q$ in US_2 as examples. Both are obtained from $\{\overline{P}Q, P\overline{Q}\}$. But $P\oplus Q$ regards $\{\overline{P}Q, P\overline{Q}\}$ as the corresponding set of the $\oplus$ operation (hence shaded), regards the two minterms not occurred in $\{\overline{P}Q, P\overline{Q}\}$ as the complement set of the $\oplus$ operation (hence blank). While ${\sim}P=^{-1}Q$

regards $\{\overline{P}Q, P\overline{Q}\}$ as the existent minterms (hence drawn), regards the two minterms not occurred in $\{\overline{P}Q, P\overline{Q}\}$ as the nonexistent minterms (hence not drawn).

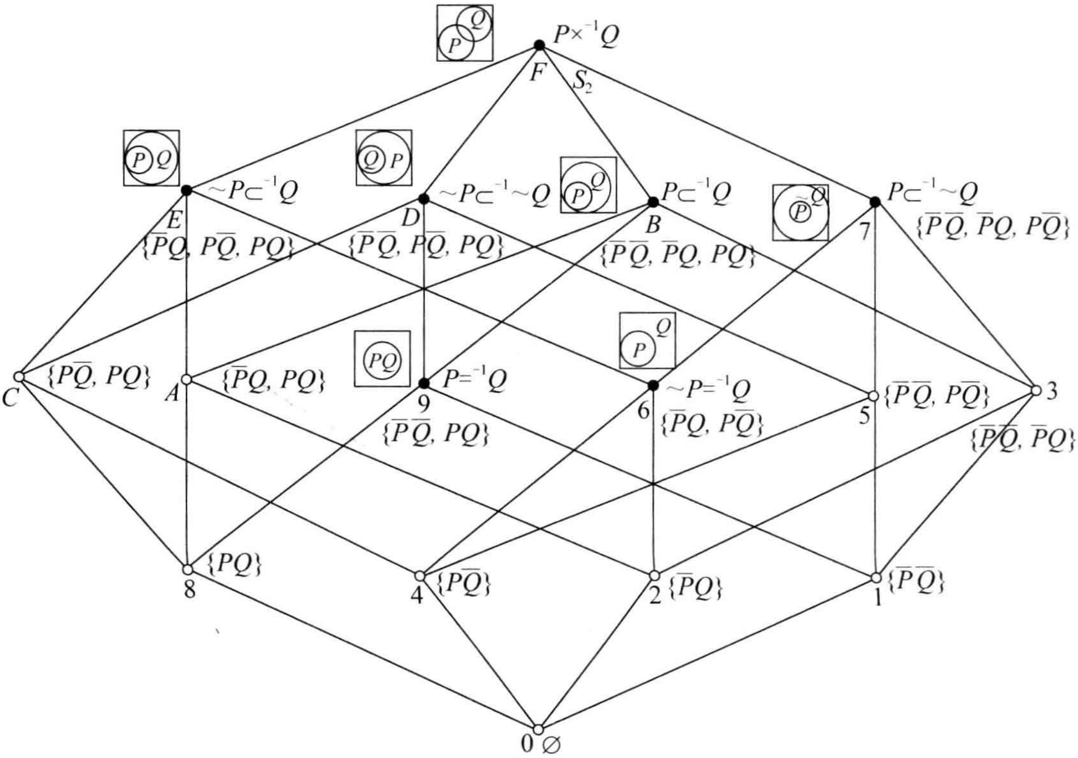


Fig. 23.11 Hasse diagram for US_2

In Fig. 23.11, the concrete vertices are real vertices, each of which represents a unicellular second-order single empirical or mathematical connection proposition with its mutually inverse diagram nearby. The hollow vertices are imaginary vertices representing meaningless second-order single empirical or mathematical connection propositions, because at least one of P or Q is a distinguished proposition. The vertices are numbered, from 0 to F. The vertices numbered C, A, 5, 3 are called characteristic vertices and have the following properties: if we take these characteristic vertices as the top vertices, take the vertex numbered 0 as the bottom vertex, and make substructures (they are not sub-Boolean algebras of Fig. 23.11, so we call them substructures instead), then all vertices in these substructures are imaginary ones; others are all real vertices.

23.3.2.2 MS_2 and logical squares

Taking every vertex in US_2 as an atom to construct Boolean algebra, we obtain the multicellular main-auxiliary algebra for single set theorems with P and Q two fact propositions $MS_2=\langle\rho(\rho(S_2)), \sim, \cap, \cup, \varnothing, \rho(S_2)\rangle$, a sketch of which is shown in Fig.

23.12, where the partially ordered relation is $\subseteq^{-1}$. In Fig. 23.12, only the real vertices are developed, and only the useful second-order single empirical or mathematical connection propositions are given.

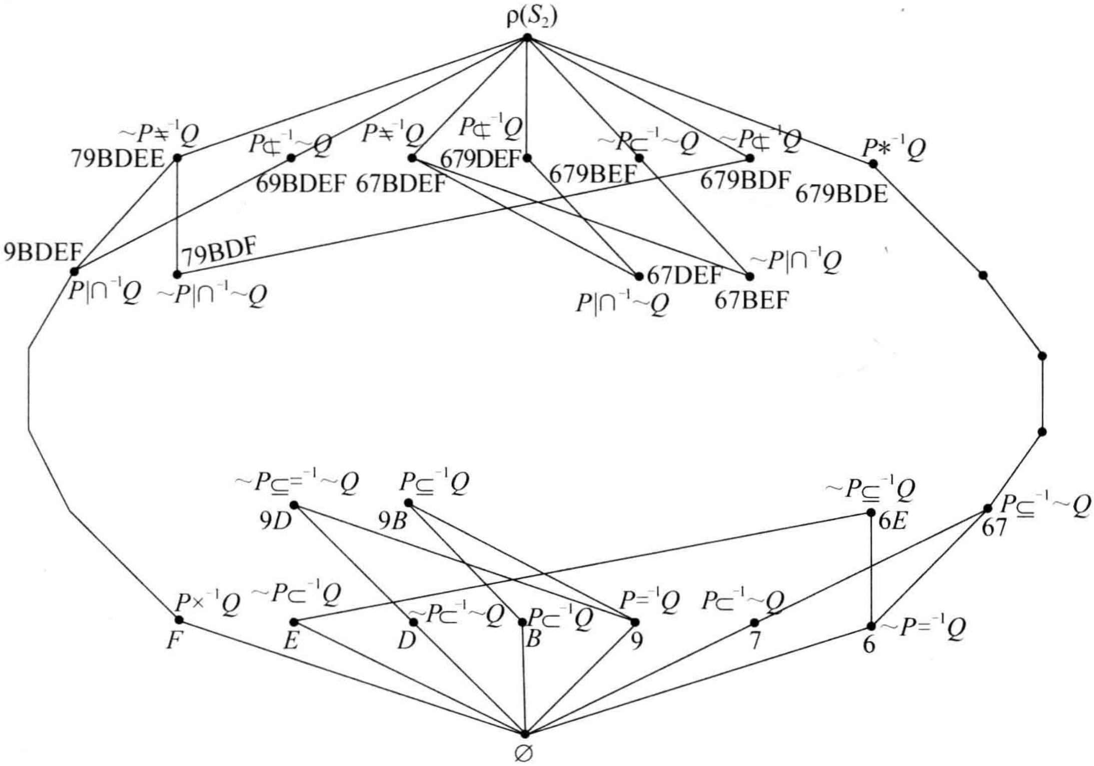


Fig. 23.12 Hasse diagram for MS_2

Similar to the construction of MS_2, we can construct MS_3 and MS_4.

MS_n can be used to prove single set theorems. Take MS_2 as an example. The proof method is as follows: suppose A, B, and C are any real vertices on MS_2; i.e., the given vertices on Fig. 23.12 except the bottom one and the top one. If B is reachable from A via an ascending path, then $A\subseteq^{-1}B$ is a single set theorem; if the path is zero path (i.e., A and B are the same vertex), then $A=^{-1}B$ is a single set theorem; otherwise, $A\subset^{-1}B$ is a single set theorem; if B is the complement of A, then A and B are contradictory propositions and $\sim A=^{-1}B$ is a single set theorem; if B is the complement of A, and B is reachable from C via an ascending path, then A and C are contrary propositions, and $\sim\{A|\cap^{-1}C\}$ or $\sim A\cup|^{-1}\sim C$ is a single set theorem; if B is the complement of A, and C is reachable from B via an ascending path, then A and C are subcontrary propositions, and $A\cup|^{-1}C$ is a single set theorem. For example, from Fig. 23.12 we learn that, $P\subseteq^{-1}Q$ is mutually inversely contained in $P|\cap^{-1}Q$, $P\subseteq^{-1}Q$ and $P|\cap^{-1}\sim Q$ are contradictory propositions, $P\subseteq^{-1}Q$ and $P\subseteq^{-1}\sim Q$ are contrary propositions, $P|\cap^{-1}Q$ and $P|\cap^{-1}\sim Q$ are subcontrary propositions. According to MS_2, we can construct the logical square shown in Fig. 23.13.

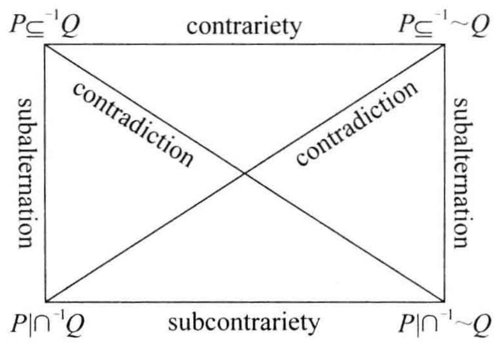


Fig. 23.13 Logical square

Similar to Q_1, we can prove that MS_n is both consistent and complete.

23.3.2.3 Correspondence between MS_2 and US_2

MS_2, MS_3, and MS_4 are theoretically stringent but not practical; first, because they have too many vertices, MS_2 has 65536 vertices, the numbers of vertices of MS_3 and MS_4 are astronomical figures and cannot be practically disposed of. Secondly, they have too many useless vertices. For example, MS_2 has 65536 vertices, but only two dozen shown in Fig. 23.12 are useful. Therefore, in single set theorem proving, what we actually use is US_n not MS_n.

How do we perform single set theorem proving on US_n? We already know how to do it on MS_n. The task remaining is to make a one-to-one correspondence between MS_n and US_n. Take MS_2 and US_2 as an example. A vertex on MS_2 is a multicellular second-order single empirical or mathematical connection proposition, a vertex on US_2 is a unicellular second-order single empirical or mathematical connection proposition, therefore, a vertex on MS_2 corresponds a set of real vertices on US_2. For example, vertex $P|\cap^{-1}Q$ on MS_2 corresponds to the set composed of all the real vertices of the substructure on US_2 with 8 being the bottom vertex and F being the top one, the common characteristic of which is the presence of the minterm PQ. Vertex $P\subseteq^{-1}Q$ (equivalent to $\sim P\cup|^{-1}Q$) on MS_2 corresponds to the set composed of all the real vertices of the substructure on US_2 with B being the top vertex and 0 being the bottom one, the common characteristic of which is the absence of the minterm $P\bar{Q}$. The greatest lower bound and least upper bound of the two vertices A and B on MS_2 correspond to the intersection and union of the two sets of real vertices A and B on US_2. For example, the greatest lower bound of the vertices $P\subseteq^{-1}Q$ and $\sim P\subseteq^{-1}\sim Q$ on MS_2 is $P=^{-1}Q$, and the intersection of the sets $P\subseteq^{-1}Q$ (composed of the real vertices 9 and B) and $\sim P\subseteq^{-1}\sim Q$ (composed of the real vertices 9 and D) on US_2 is the set composed of vertex $P=^{-1}Q$ (vertex 9). Vertex B being reachable from vertex A via an ascending path on MS_2 corresponds to set A being contained in set B on US_2. For example, on MS_2, vertex $P|\cap^{-1}Q$

is reachable from vertex $P\subseteq^{-1}Q$ via an ascending path, and on US_2, set $P\subseteq^{-1}Q$ (composed of 9 and B) is contained in set $P|\cap^{-1}Q$ (composed of 9, B, D, E, and F). Vertex A and vertex B being the same vertex on MS_2 corresponds to set A and set B being equivalent on US_2. For example, on MS_2, vertex $\{P\subseteq^{-1}Q\}\cap\{\sim P\subseteq^{-1}\sim Q\}$ and vertex $P=^{-1}Q$ are the same, and on US_2, set $\{P\subseteq^{-1}Q\}\cap\{\sim P\subseteq^{-1}\sim Q\}$ and set $P=^{-1}Q$ are equivalent. Vertex B being reachable from vertex A via a nonzero ascending path on MS_2 corresponds to set A being properly contained in set B on US_2. For example, on MS_2, vertex $P|\cap^{-1}Q$ is reachable from vertex $P\subseteq^{-1}Q$ via a nonzero ascending path, and on US_2, set $P\subseteq^{-1}Q$ (composed of 9 and B) is properly contained in set $P|\cap^{-1}Q$ (composed of 9, B, D, E, and F). Vertex B being the complement (contradictory proposition) of vertex A on MS_2 corresponds to set B being the complement set of set A on US_2; i.e., their union is the universal set, their intersection is the empty set. For example, on MS_2, vertex $P\subseteq^{-1}\sim Q$ is the complement of vertex $P|\cap^{-1}Q$, and on US_2, set $P\subseteq^{-1}\sim Q$ (composed of 6 and 7) is the complement set of set $P|\cap^{-1}Q$ (composed of 9, B, D, E, and F), their union is the universal set, their intersection is the empty set. Vertices A and C being contrary propositions on MS_2 corresponds to the union of sets A and C not being the universal set, the intersection of A and C being the empty set on US_2. For example, on MS_2, $P\subseteq^{-1}Q$ and $P\subseteq^{-1}\sim Q$ are contrary propositions, and on US_2, the union of set $P\subseteq^{-1}Q$ (composed of 9 and B) and set $P\subseteq^{-1}\sim Q$ (composed of 6 and 7) is not the universal set, the intersection of them is the empty set. Vertices A and C being subcontrary propositions on MS_2 corresponds to the union of sets A and C being the universal set, the intersection of A and C not being the empty set on US_2. For example, on MS_2, $P|\cap^{-1}Q$ and $P|\cap^{-1}\sim Q$ are subcontrary propositions, and on US_2, the union of set $P|\cap^{-1}Q$ (composed of 9, B, D, E, and F) and set $P|\cap^{-1}\sim Q$ (composed of 6, 7, D, E, and F) is the universal set, their intersection is not the empty set. The bottom vertex and top vertex on MS_2 correspond to the empty set and the universal set on US_2.

According to the above correspondence between MS_2 and US_2, US_2 can be used to prove single set theorems as follows: suppose A and B are two sets of real vertices on US_2, and A and B are neither empty set nor universal set. If A is contained in B, then $A\subseteq^{-1}B$ is a single set theorem. If A and B are equivalent, then $A=^{-1}B$ is a single set theorem. If A is properly contained in B, then $A\subset^{-1}B$ is a single set theorem. If B is the complement set of A, then $\sim A=^{-1}B$ is a single set theorem. If A and B are contrary propositions; i.e., $A\cup B$ is not the universal set, and $A\cap B$ is the empty set, then $\sim A\cup|^{-1}\sim B$ is a single set theorem. If A and B are subcontrary propositions; i.e., $A\cup B$ is the universal set, and $A\cap B$ is not the empty set, then $A\cup|^{-1}B$ is a single set theorem.

23.3.2.4 Logical rectangles and logical pies

According to US_2, we can make the logical rectangle shown in Fig. 23.14. In Fig.

23.14, the rectangle denotes the universal set {6, 7, 9, B, D, E, F}. The two vertical dotted lines divide the rectangle into left, middle, and right three compartments. The left compartment {9, B} denotes $P\subseteq^{-1}Q$, the right compartment {6, 7} denotes $P\subseteq^{-1}\sim Q$, the left-middle compartments {9, B, D, E, F} denote $P|\cap^{-1}Q$, the middle-right compartments {6, 7, D, E, F} denote $P|\cap^{-1}\sim Q$. All the opposition relations in Fig. 23.13 correspond to something in Fig. 23.14. For example, the subalternation of $P\subseteq^{-1}Q$ and $P|\cap^{-1}Q$ in Fig. 23.13 corresponds to $P\subseteq^{-1}Q$ being contained in $P|\cap^{-1}Q$ in Fig. 23.14. The contradiction of $P\subseteq^{-1}Q$ and $P|\cap^{-1}\sim Q$ in Fig. 23.13 corresponds to $P|\cap^{-1}\sim Q$ (the middle-right compartments) being the complement set of $P\subseteq^{-1}Q$ (the left compartment) in Fig.23.14. The contrariety of $P\subseteq^{-1}Q$ and $P\subseteq^{-1}\sim Q$ in Fig. 23.13 corresponds to the union of $P\subseteq^{-1}Q$ (the left compartment) and $P\subseteq^{-1}\sim Q$ (the right compartment) not being the universal set, the intersection being the empty set in Fig. 23.14. The subcontrariety of $P|\cap^{-1}Q$ and $P|\cap^{-1}\sim Q$ in Fig. 23.13 corresponds to the union of $P|\cap^{-1}Q$ (the left-middle compartments) and $P|\cap^{-1}\sim Q$ (the middle-right compartments) being the universal set, the intersection not being the empty set. Fig. 23.14 also reveals some information that Fig. 23.13 lack: the middle compartment is one that both the contrary propositions $P\subseteq^{-1}Q$ and $P\subseteq^{-1}\sim Q$ not cover and the subcontrary propositions $P|\cap^{-1}Q$ and $P|\cap^{-1}\sim Q$ overlap.

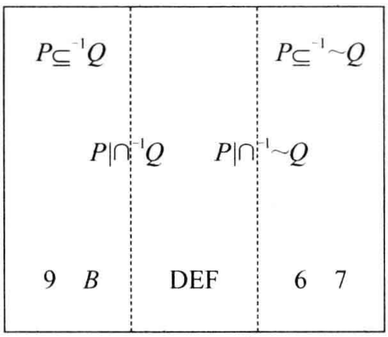


Fig. 23.14 Logical rectangle

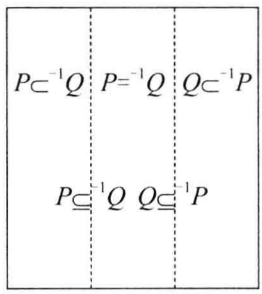


Fig.23.15 Logical rectangle of 5 propositions

In some logical rectangles, the fifth proposition can be put into the middle compartment to form a logical rectangle of five propositions, as shown in Fig. 23.15. Fig. 23.15 can denotes the six opposition relations that a logical square can denotes: the subalternation of $P\subset^{-1}Q$ and $P\subseteq^{-1}Q$, the subalternation of $Q\subset^{-1}P$ and $Q\subseteq^{-1}P$, the contradiction of $P\subset^{-1}Q$ and $Q\subseteq^{-1}P$, the contradiction of $Q\subset^{-1}P$ and $P\subseteq^{-1}Q$, the contrariety of $P\subset^{-1}Q$ and $Q\subset^{-1}P$, the subcontrariety of $P\subseteq^{-1}Q$ and $Q\subseteq^{-1}P$. Fig. 23.15 can also denotes four opposition relations that a logical square cannot denote: the contrariety of $P=^{-1}Q$ and $P\subset^{-1}Q$, the contrariety of $P=^{-1}Q$ and $Q\subset^{-1}P$, the subalternation of $P=^{-1}Q$ and $P\subseteq^{-1}Q$, the subalternation of $P=^{-1}Q$ and $Q\subseteq^{-1}P$.

The left compartment and the right compartment of Fig. 23.15 are not adjacent, not to form the left-right compartments. Whereas, the left-right compartments can denote the proposition $P\neq^{-1}Q$. Therefore, we connect the left compartment and the right compartment to form the logical pie shown in Fig. 23.16.

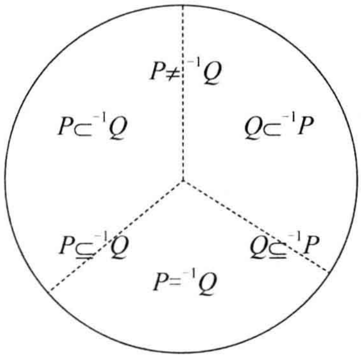


Fig. 23.16 Logical pie

In addition to the 10 opposition relations Fig. 23.15 denote, Fig. 23.16 can also denotes the 5 opposition relation that Fig. 23.15 cannot denote: the subalernation of $P\subset^{-1}Q$ and $P\neq^{-1}Q$, the subalternation of $Q\subset^{-1}P$ and $P\neq^{-1}Q$, the contradiction of $P=^{-1}Q$ and $P\neq^{-1}Q$, the subcontrariety of $P\subseteq^{-1}Q$ and $P\neq^{-1}Q$, the subcontrariety of $Q\subseteq^{-1}P$ and $P\neq^{-1}Q$.

23.3.3 Main-auxiliary algebras of multiple set theorems

Eliminating the bottom propositions of Figs. 5.33 and 5.34, merging the two next to the bottom propositions into one, reversing top side down, then we obtain the main-auxiliary algebra for multiple set theorems with $p(x)$, $q(y)$, and $r(x, y)$ three fact propositions M_3, as shown in Fig. 23.17. The vertices in Fig. 23.17 are numbered, near each vertex, its least success diagram is given.

M_3 is a complemented lattice. The complements of each element are not unique. Suppose A is any element in M_3, A has more than one complements, and $\sim A$ is its main complement. For example, vertex 2 has two complements: vertices 3 and 5, and vertex 5 is its main complement.

Multiple set theorem proving on M_3 is performed as follows: suppose A, B, and C are any vertices on M_3 (each vertex is a property proposition segment). If B is reachable from A via an ascending path, then $\mathrm{A}r(x, y)\subseteq^{-1}\mathrm{B}r(x, y)$ is a multiple set theorem. For example, vertex 4 is reachable from vertex 2 via an ascending path, hence $\{q(y)|\cap^{-1}\{p(x)\subseteq^{-1}r(x, y)\}\}\subseteq^{-1}\{p(x)\subseteq^{-1}q(y)|\cap^{-1}r(x, y)\}$ is a multiple set theorem. If vertex B is reachable from vertex A via a zero ascending path; i.e., vertices A and B are the same vertex, then $\mathrm{A}r(x, y)=^{-1}\mathrm{B}r(x, y)$ is a multiple set theorem. For example, vertex 1 is reachable from vertex 1 via

a zero ascending path, hence $\{p(x)\subseteq^{-1}q(y)\subseteq^{-1}r(x, y)\}=^{-1}\{q(y)\subseteq^{-1}p(x)\subseteq^{-1}r(x, y)\}$ is a multiple set theorem. If B is the main complement of A, then A and B are contradictory propositions, and $\sim\{\mathrm{A}r(x, y)=^{-1}\mathrm{B}\sim r(x, y)$ is a multiple set theorem. For example, vertex 5 is the main complement of vertex 2, hence $\sim\{q(y)|\cap^{-1}\{p(x)\subseteq^{-1}r(x, y)\}\}=^{-1}\{q(y)\subseteq^{-1}p(x)|\cap^{-1}\sim r(x, y)\}$ is a multiple set theorem. If B is the main complement of A, and B is reachable from C via a nonzero ascending path, then $\mathrm{A}r(x, y)$ and $\mathrm{C}\sim r(x, y)$ are contrary propositions, and $\sim\mathrm{A}r(x, y)\cup|^{-1}\sim\mathrm{C}\sim r(x, y)$ is a multiple set theorem. For example, vertex 5 is the main complement of vertex 2, and vertex 5 is reachable from vertex 3 via a nonzero ascending path, hence vertices 2 and 3 are contrary propositions, and $\sim\{q(y)|\cap^{-1}\{p(x)\subseteq^{-1}r(x, y)\}\}\cup|^{-1}\sim\{p(x)|\cap^{-1}\{q(y)\subseteq^{-1}\sim r(x, y)\}\}$ is a multiple set theorem. If B is the main complement of A, and C is reachable from B via a nonzero ascending path, then $\mathrm{A}r(x, y)$ and $\mathrm{C}\sim r(x, y)$ are subcontrary propositions, and $\mathrm{A}r(x, y)\cup|^{-1}\mathrm{C}\sim r(x, y)$ is a multiple set theorem. For example, vertex 2 is the main complement of vertex 5, and vertex 4 is reachable from vertex 2 via a nonzero ascending path, hence vertices 5 and 4 are subcontrary propositions, and $\{q(y)\subseteq^{-1}p(x)|\cap^{-1}r(x, y)\}\cup|^{-1}\{p(x)\subseteq^{-1}q(y)|\cap^{-1}\sim r(x, y)\}$ is a multiple set theorem.

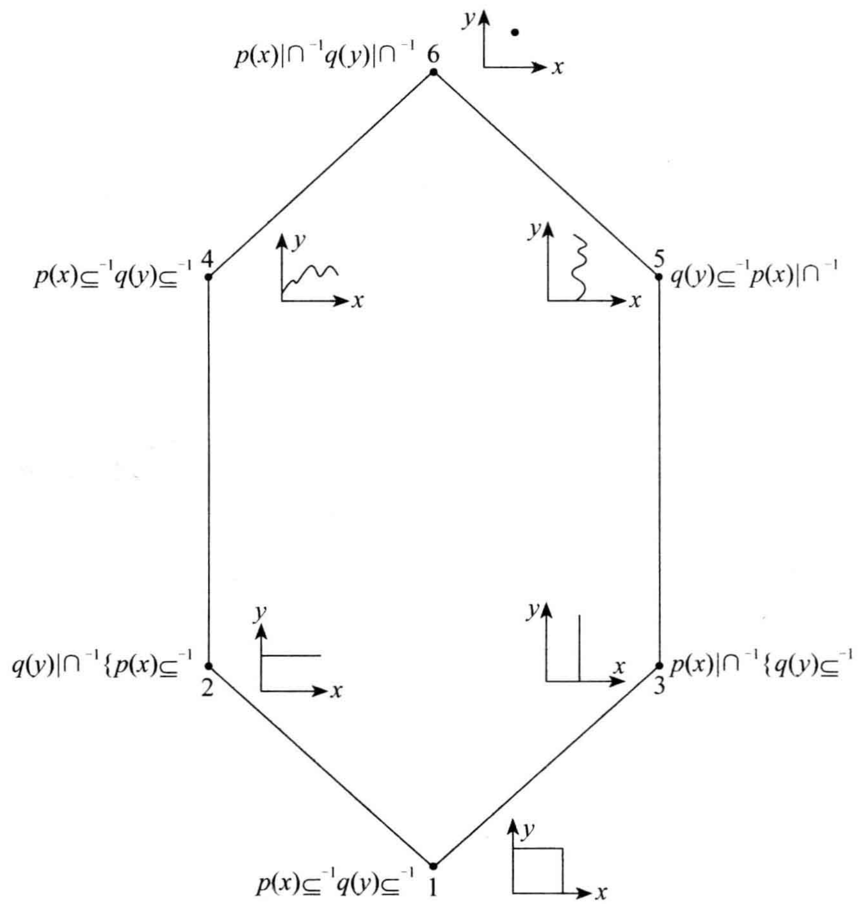


Fig. 23.17 Hasse diagram for M_3

Similar to Q_1, it can be proved that M_3 is both consistent and complete.

23.3.4 Main-auxiliary algebras for semi-set theorems and logical cube

23.3.4.1 Main-auxiliary algebras for semi-set theorems

If one side of a set connection operator is a single empirical or mathematical connection proposition, the other side is a fact proposition, then the proposition obtained is called a semi-set connection proposition. For example, $\sim P\subseteq^{-1}\{P\subset^{-1}Q\}$, $Q\subseteq^{-1}\{P\subset^{-1}Q\}$, $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$, etc. are semi-set connection propositions. The Q in $\sim P\subseteq^{-1}\{P\subset^{-1}Q\}$ is a free proposition variable, and the P in $Q\subseteq^{-1}\{P\subset^{-1}Q\}$ is a free proposition variable, therefore these implication paradoxes cannot be semi-set theorems. Both P and Q in $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$ are bound proposition variables, therefore, it can be proved to be a semi-set theorem.

Main-auxiliary algebra for semi-set theorem SE_2 is obtained by piecing US_2 and Q_2 together. It retains the seven real vertices of US_2, and the nine imaginary vertices of US_2 are replaced by the corresponding vertices of Q_2, as is shown in Fig. 23.18.

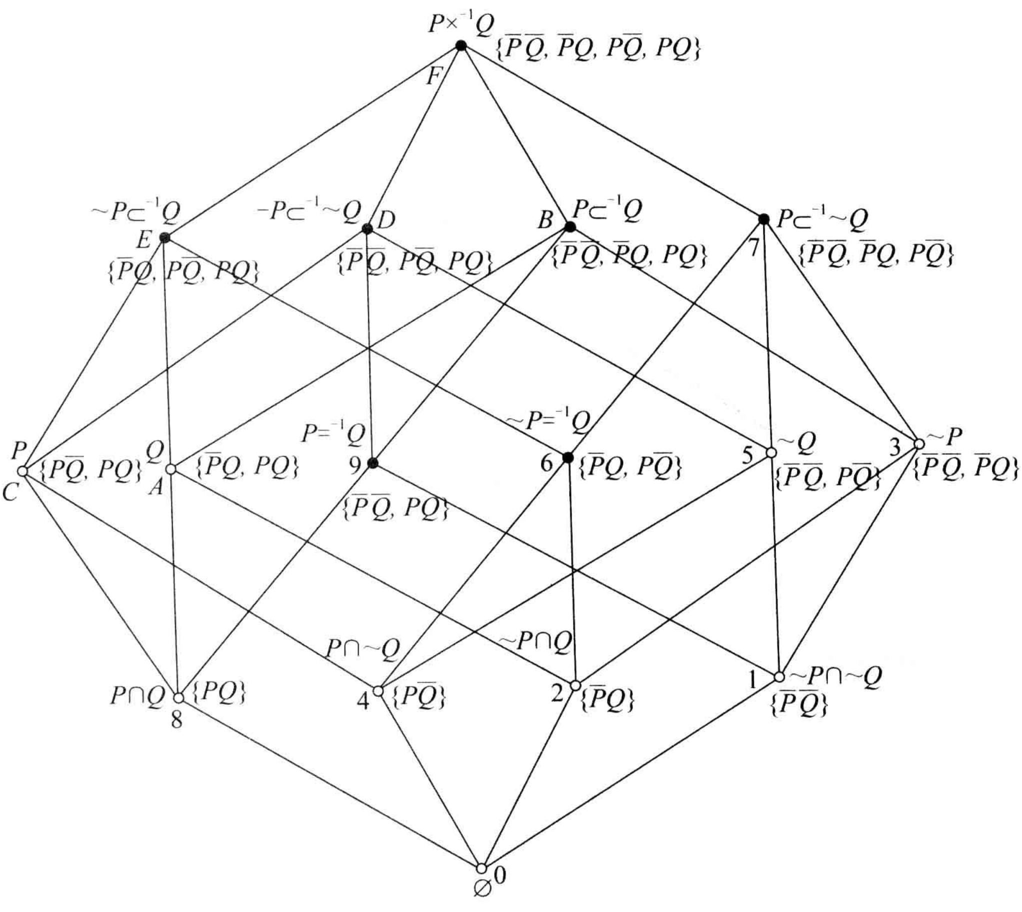


Fig. 23.18 Hasse diagram for SE_2

The set of all the real vertices of the substructure on SE_2 with vertex 8 at the bottom, vertex F at the top (i.e., vertices 9, B, D, E, and F) denotes $P|\cap^{-1}Q$. Similarly, $P|\cap^{-1}\sim Q$, $\sim P|\cap^{-1}Q$, and $\sim P|\cap^{-1}\sim Q$ can be denoted. The set of all the real vertices of the substructure on SE_2 with vertex 0 at the bottom, vertex 7 at the top (i.e., vertices 6 and 7) denotes $P\subseteq^{-1}\sim Q$ (i.e., $\sim P\cup|^{-1}\sim Q$). Similarly, $P\subseteq^{-1}Q$ (i.e., $\sim P\cup|^{-1}Q$), $\sim P\subseteq^{-1}\sim Q$ (i.e., $P\cup|^{-1}\sim Q$), and $\sim P\subseteq^{-1}Q$ (i.e., $P\cup|^{-1}Q$) can be denoted.

Using SE_2 we can prove single set theorems $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$, $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{\sim P|\cap^{-1}\sim Q\}$, $\sim\{P|\cap^{-1}Q\}=^{-1}\{P\subseteq^{-1}\sim Q\}$, $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}P\}=^{-1}\{P=^{-1}Q\}$, etc., the proof method is the same as US_2. Using SE_2 we can prove quasi-set theorems $P\cap Q\subseteq^{-1}P$, $P\cap Q\subseteq^{-1}Q$, etc., the proof method is the same as Q_2. Using SE_2 we can prove semi-set theorems $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$ (proof by refutation), $\{P\subset^{-1}Q\}\cap\{P\subset^{-1}\sim Q\}=^{-1}\sim P$ (reduction to absurdity), $\{P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}Q\}=^{-1}Q$, and $\{P\subset^{-1}\sim Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}\sim Q$. Take $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$ (proof by refutation) as an example. It is proved as follows: the greatest lower bound of vertex E (denoting $\sim P\subset^{-1}Q$) and vertex D (denoting $\sim P\subset^{-1}\sim Q$) is vertex C (denoting P), therefore $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$ is a semi-set theorem.

23.3.4.2 Logical cube

The shape of the logical cube is like the Rubic cube, it is also composed of 3∗3∗3 cubes. Every cube of the logical cube denotes a vertex of SE_2. There are 27 cubes in the logical cube, whereas there are only 16 vertices in SE_2. Therefore, for some vertices of SE_2, one vertex corresponds to several cubes of the logical cube. A cube of the logical cube is shown in Fig. 23.19.

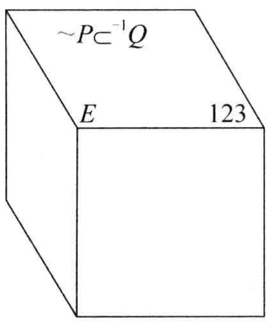


Fig. 23.19 A cube of the logical cube

In Fig. 23.19, at the top is the proposition $\sim P\subset^{-1}Q$ the cube denotes, at the bottom left corner is the number E of the vertex on SE_2 the proposition $\sim P\subset^{-1}Q$ corresponds, at the bottom right corner is the position the cube situates: the first level, the second row, the third column.

The logical cube is shown in Fig. 23.20.

In Fig. 23.21, the top left side of the double line is the proposition $P|\cap^{-1}Q$, the bottom right side is the proposition $P\subseteq^{-1}\sim Q$ (Here we deem that $\varnothing$ can occur in both $P|\cap^{-1}Q$ and

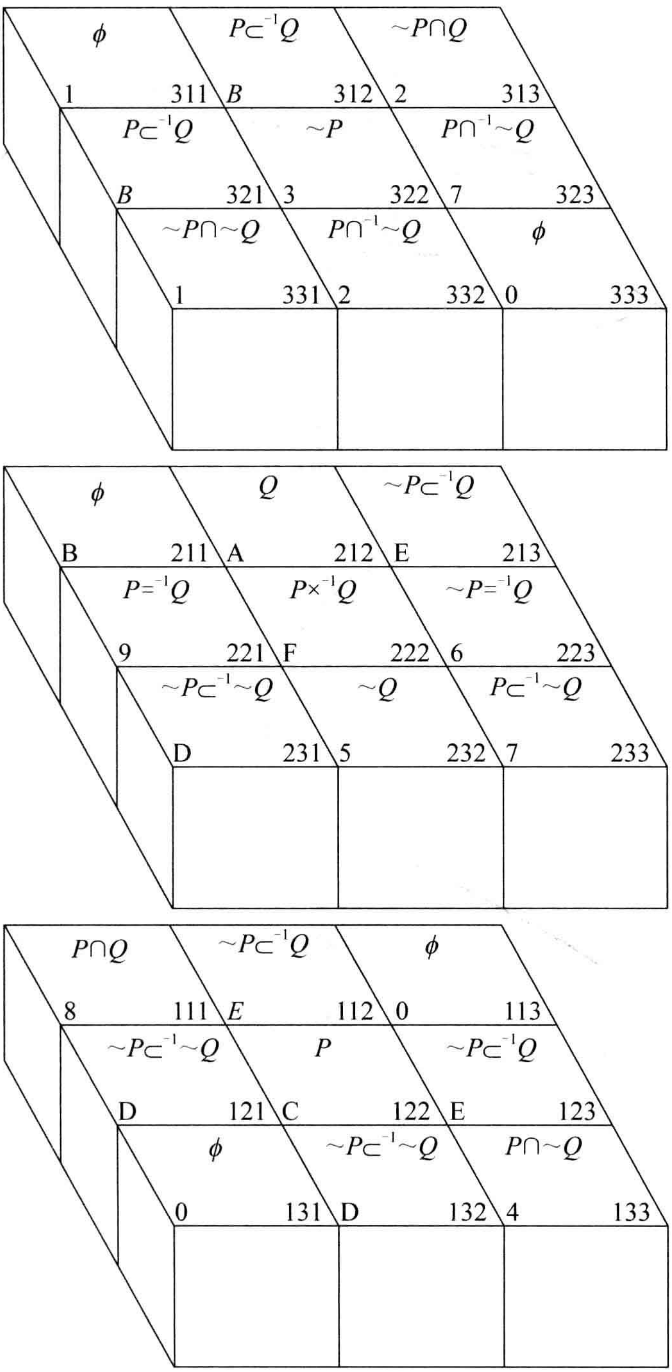


Fig. 23.20 Logical cube

$P\subseteq^{-1}\sim Q$). Fig. 23.21 can be used to denote $\sim\{P|\cap^{-1}Q\}=^{-1}\{P\subseteq^{-1}\sim Q\}$: the union of $P|\cap^{-1}Q$ and $P\subseteq^{-1}\sim Q$ is the whole logical cube, the intersection is the empty set. Fig. 23.21 can also be used to denote $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$: $P\subseteq^{-1}Q$ is composed of cubes 221 (vertex 9), 211 (vertex B), 312 (vertex B), and 321 (vertex B), they are contained in $P|\cap^{-1}Q$.

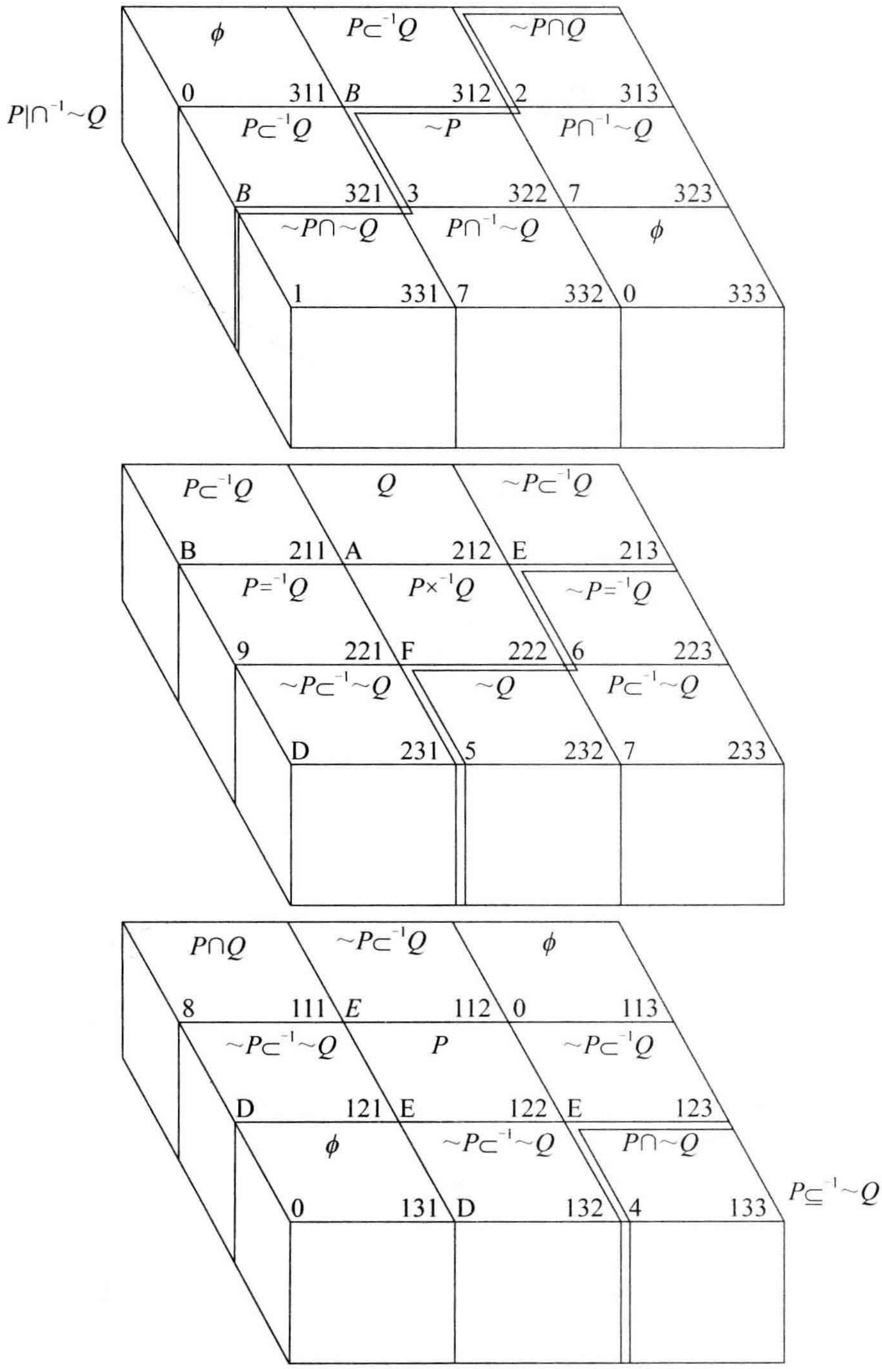


Fig. 23.21 $\sim\{P|\cap^{-1}Q\}=^{-1}\{P\subseteq^{-1}\sim Q\}$ **and** $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$

In Fig. 23.22, the bottom left side of the double line is the proposition $\sim P|\cap^{-1}\sim Q$. Fig. 23.22 can be used to denote $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{\sim P|\cap^{-1}\sim Q\}$: $P\subseteq^{-1}Q$ is composed of cubes 221 (vertex 9), 211 (vertex B), 312 (vertex B), and 321 (vertex B), they are contained in $\sim P|\cap^{-1}\sim Q$.

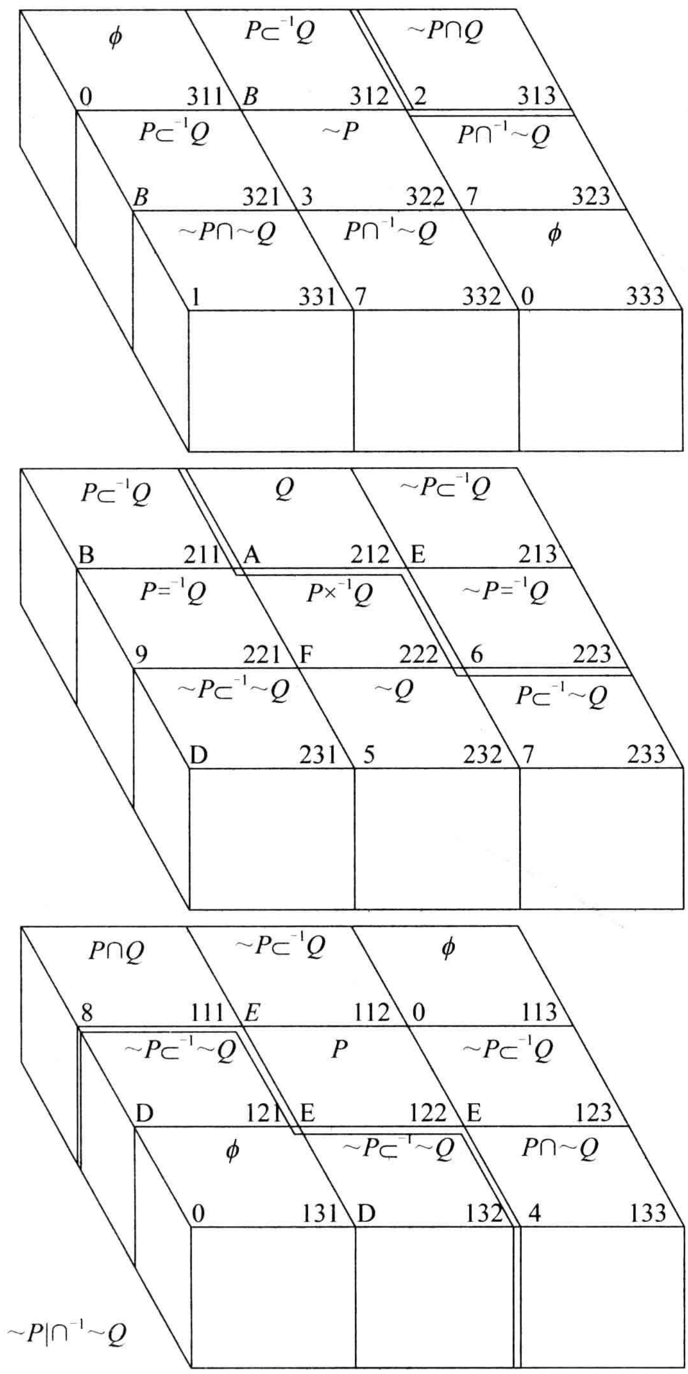


Fig. 23.22 $\{P\subseteq^{-1}Q\}\subseteq^{-1}\{\sim P|\cap^{-1}\sim Q\}$

In SE_2 (Fig. 23.18), the vertices $P|\cap^{-1}Q$ and $P|\cap^{-1}\sim Q$ overlap are C, D, E, and F, the vertices they do not cover are 0, 1, 2, and 3. In Fig. 23.23, the top left side of the dotted line denotes $P|\cap^{-1}Q$, the bottom right side of the concrete line denotes $P|\cap^{-1}\sim Q$. The vertices they overlap are 0, C, D, E, and F, the vertices they do not cover are 1, 2, and 3.

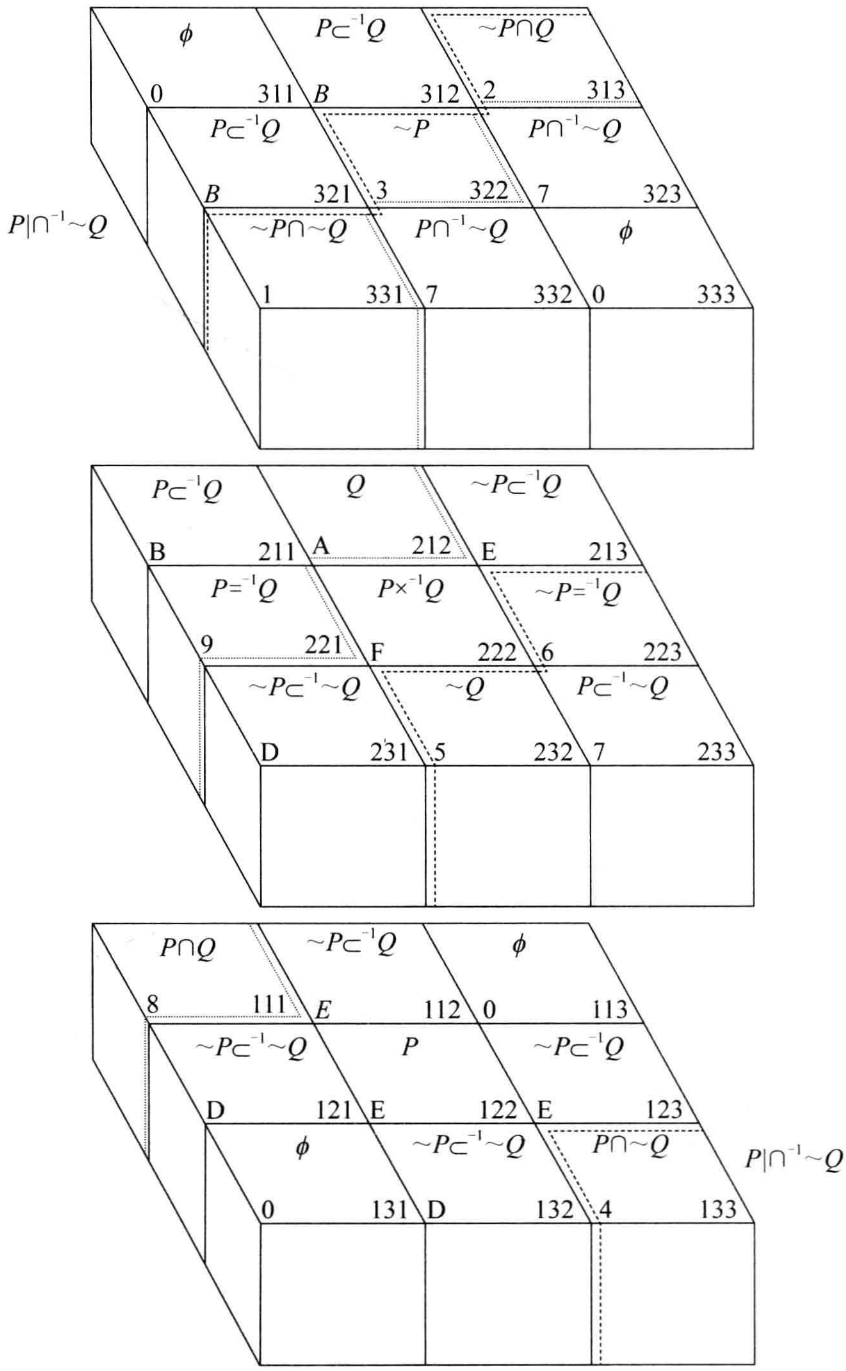


Fig. 23.23 $P|\cap^{-1}Q$ **and** $P|\cap^{-1}\sim Q$

In the logical cube, single set theorem $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}P\}=^{-1}\{P=^{-1}Q\}$ is denoted by the logical rectangle shown in Fig. 23. 24.

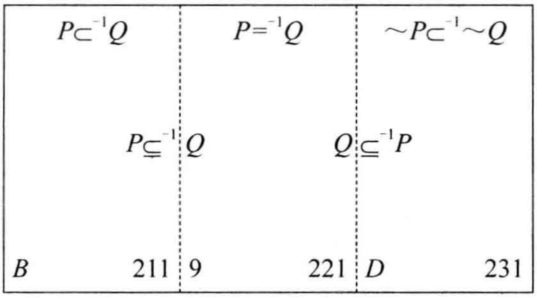


Fig. 23.24 $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}P\}=^{-1}\{P=^{-1}Q\}$

In the logical cube, semi-set theorem $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$ is denoted by the logical rectangle shown in Fig. 23.25.

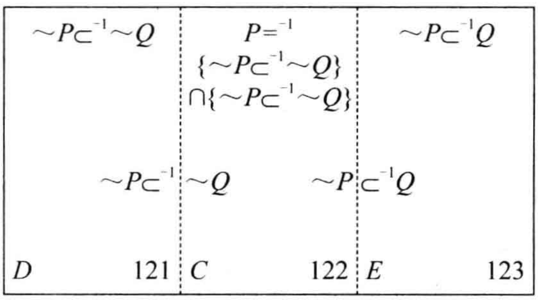


Fig. 23. 25 $\{\sim P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}P$

In the logical cube, semi-set theorem $\{P\subset^{-1}Q\}\cap\{P\subset^{-1}\sim Q\}=^{-1}\sim P$ is denoted by the logical rectangle shown in Fig. 23.26.

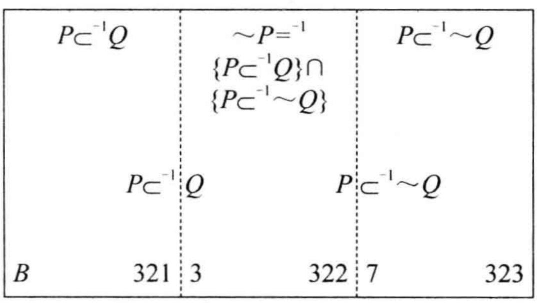


Fig. 23.26 $\{P\subset^{-1}Q\}\cap\{P\subset^{-1}\sim Q\}=^{-1}\sim P$

In the logical cube, semi-set theorem $\{P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}Q\}=^{-1}Q$ is denoted by the logical rectangle shown in Fig. 23.27.

$\sim P\subset^{-1}Q$	$Q=^{-1}$ $\{\sim P\subset^{-1}Q\}$ $\cap\{P\subset^{-1}Q\}$	$P\subset^{-1}Q$
$\sim P\subset^{-1}$	Q $\quad$ P	$\subset^{-1}Q$
E $\quad$ 112	A $\quad$ 212	B $\quad$ 312

Fig. 23.27 $\{P\subset^{-1}Q\}\cap\{\sim P\subset^{-1}Q\}=^{-1}Q$

In the logical cube, semi-set theorem $\{P\subset^{-1}\sim Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}\sim Q$ is denoted by the logical rectangle shown in Fig. 23.28.

$\sim P\subset^{-1}\sim Q$	$\sim Q=^{-1}$ $\{\sim P\subset^{-1}\sim Q\}$ $\cap\{P\subset^{-1}\sim Q\}$	$P\subset^{-1}\sim Q$
$\sim P\subset^{-1}$	$\sim Q$ $\quad$ P	$\subset^{-1}\sim Q$
D $\quad$ 132	5 $\quad$ 232	7 $\quad$ 332

Fig. 23.28 $\{P\subset^{-1}\sim Q\}\cap\{\sim P\subset^{-1}\sim Q\}=^{-1}\sim Q$

23.3.5 Main-auxiliary algebras vs. Boolean algebras and complemented lattice

Q_1 and Q_2 differ from Boolean algebra in that $C\subseteq^{-1}D$ in Q_1 and Q_2 requires that C and D are not distinguished sets, but in Boolean algebra there is no such requirement. MS_n differs from Boolean algebra in that the sub-algebra of MS_n may not be an MS_n because it may not be used to prove single set theorems. US_n is not a Boolean algebra at all because the partially ordered relation $\subseteq^{-1}$ does not exist. M_3 differs from a complemented lattice in that the former has the concept of a main complement while the latter does not.

23.3.6 Decision problems

A decision problem refers to the problem that given a set proposition, if there exists an algorithm for a main-auxiliary algebra for set theorems that can decide in finite steps whether the proposition is a set theorem or not. If such an algorithm exists, then the main-auxiliary algebra for set theorems is decidable; otherwise, it is undecidable.

All the main-auxiliary algebras for set theorems in this chapter are decidable. Because algorithms are found for them, and are implemented on personal computers as provers. For any inputted set proposition, the provers can decide within seconds whether the proposition is a set theorem or not. For the algorithms, see Chapter 33.

Part 10 Universal matrices

Universal matrices unify planar matrices, multi-dimensional matrices, scalars, vectors, and tensors. Isodimensional (isodimensional is a new English word coined by the author) matrices and tensor matrices are the special cases of universal matrices. Isodimensional matrices refer to such matrices that an n-dimensional matrix can only be multiplied by an n-dimensional matrix, and the matrix yield is also an n-dimensional matrix. In this chapter 3-isodimensional matrices; i.e., cuboidal matrices are discussed in detail. Tensor matrices refer to that an nth-order tensor is denoted by an n-dimensional matrix, so that tensor algebra becomes tensor matrix algebra.

An m-dimensional matrix multiplied by an n-dimensional matrix, the resulting matrix is any dimension from $|m-n|$ to $m+n$.

Chapter 24
Universal matrices

24.1 Representation of n-dimensional matrices

24.1.1 Representation of 0-dimensional matrices

A 0-dimensional matrix is a scalar denoted by: a.

24.1.2 Representation of 1-dimensional matrices

A 1-dimensional matrix is a vector denoted by a row shown in Fig. 24.1 or a column shown in Fig. 24.2.

$$\begin{bmatrix} a_1, a_2, \ldots, a_n \end{bmatrix}$$

Fig. 24.1 Vector

$$\begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}$$

Fig. 24.2 Vector

The subscripts n in these vectors are the order of the vectors.

24.1.3 Representation of 2-dimensional matrices

A 2-dimensional matrix is a planar matrix denoted by Fig. 24.3.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Fig. 24.3 Planar matrix

Fig. 24.3 is an m×n order matrix. An n×n order planar matrix is called a square matrix.

24.1.4 Representation of 3-dimensional matrices

A 3-dimensional matrix is a cuboidal matrix whose three dimensions are: height *h*, width *w*, and depth *d* shown in Fig. 24.4.

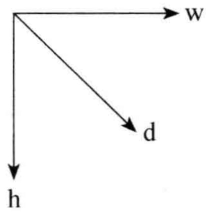


Fig. 24.4 Three dimensions of a cuboidal matrix

Cuboidal matrix A is usually denoted by A_{pqr} or $A=[a_{ijk}]_{pqr}$, in which a_{ijk} stands for an element of the *i*th height, *j*th width, and *k*th depth, p, q, and r stand for the orders of the height, width, and depth of A respectively. A can be denoted by a number of planar matrices A_{Hi} (i=1, 2, …, p) along the height direction. Each planar matrix in it is called a height plane, and A_{Hi} is the *i*th height plane. This representation is called the height plane representation, and is shown in Fig. 24.5.

$$\begin{array}{cccc} a_{111} & a_{121} & \cdots & a_{1q1} \\ a_{112} & a_{122} & \cdots & a_{1q2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{11r} & a_{12r} & \cdots & a_{1qr} \end{array}$$

$$\begin{array}{cccc} a_{211} & a_{221} & \cdots & a_{2q1} \\ a_{212} & a_{222} & \cdots & a_{2q2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{21r} & a_{22r} & \cdots & a_{2qr} \end{array}$$

$$\vdots$$

$$\begin{array}{cccc} a_{p11} & a_{p21} & \cdots & a_{pq1} \\ a_{p12} & a_{p22} & \cdots & a_{pq2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p1r} & a_{p2r} & \cdots & a_{pqr} \end{array}$$

Fig. 24.5 Height plane representation

Similarly, A can be denoted by a number of width planes A_{Wj} (j=1, 2, …, q) called the width plane representation shown in Fig. 24.6.

$$
\left[\begin{array}{cccc}
a_{111} & & & \\
a_{211} & a_{112} & & \\
\cdot & a_{212} & \cdot & \\
a_{p11} & \cdot & \cdot & a_{114} \\
 & a_{p12} & \cdot & a_{21r} \\
 & & \cdot & \cdot \\
 & & & a_{p1r}
\end{array}\right]
\quad
\left[\begin{array}{cccc}
a_{121} & & & \\
a_{221} & a_{122} & & \\
\cdot & a_{222} & \cdot & \\
a_{p21} & \cdot & \cdot & a_{12r} \\
 & a_{p22} & \cdot & a_{22r} \\
 & & \cdot & \cdot \\
 & & & a_{p2r}
\end{array}\right]
\quad
\left[\begin{array}{cccc}
a_{1q1} & & & \\
a_{2q1} & a_{1q2} & & \\
\cdot & a_{2q2} & \cdot & \\
a_{pq1} & \cdot & \cdot & a_{1qr} \\
 & a_{pq2} & \cdot & a_{2qr} \\
 & & \cdot & \cdot \\
 & & & a_{pqr}
\end{array}\right]
$$

Fig. 24.6 Width plane representation

A can also be denoted by a number of depth planes A_{Dk} (k=1, 2, …, r) called the depth plane representation shown in Fig. 24.7.

$$
\begin{bmatrix}
a_{111} & a_{121} & \cdots & a_{1q1} \\
a_{211} & a_{221} & \cdots & a_{2q1} \\
\cdots & \cdots & \cdots & \cdots \\
a_{p11} & a_{p21} & \cdots & a_{pq1}
\end{bmatrix}
$$

$$
\begin{bmatrix}
a_{112} & a_{122} & \cdots & a_{1q2} \\
a_{212} & a_{222} & \cdots & a_{2q2} \\
\cdots & \cdots & \cdots & \cdots \\
a_{p12} & a_{p22} & \cdots & a_{pq2}
\end{bmatrix}
$$

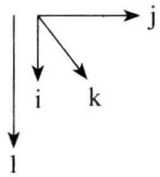
$$\vdots$$

$$
\begin{bmatrix}
a_{11r} & a_{12r} & \cdots & a_{1qr} \\
a_{21r} & a_{22r} & \cdots & a_{2qr} \\
\cdots & \cdots & \cdots & \cdots \\
a_{p1r} & a_{p2r} & \cdots & a_{pqr}
\end{bmatrix}
$$

Fig. 24.7 Depth plane representation

24.1.5 Representation of *n*-dimensional matrices with *n*>3

An example of the depth plane representation of a 4-dimensional matrix is shown in Fig. 24.8.



$$
\left\{
\begin{bmatrix} a_{1111} & a_{1211} \\ a_{2111} & a_{2211} \end{bmatrix}
\quad
\begin{bmatrix} a_{1121} & a_{1212} \\ a_{2121} & a_{2221} \end{bmatrix}
\right\}
$$

$$
\left\{
\begin{bmatrix} a_{1112} & a_{1212} \\ a_{2112} & a_{2212} \end{bmatrix}
\quad
\begin{bmatrix} a_{1122} & a_{1222} \\ a_{2122} & a_{2222} \end{bmatrix}
\right\}
$$

Fig. 24.8 A 4-dimensional matrix

24.2 Multiplications

24.2.1 Inner products

Definition 24.1: Suppose $A=[a_{ij}]_{pq}$ is a 2-dimensional matrix, $B=[b_j]_q$ is a 1-dimensional matrix, A and B have one index, say *j*, in common, and both orders of *j* are q, then $C=[c_i]_p=[\sum_{j=1}^{q} a_{ij} * b_j]_p$, where $*$ is the ordinary multiplication, is called the inner product of A and B, denoted by $C=A*_iB$. C is a 1-dimensional matrix.

It is easy to generalize this definition to the inner product of an m-dimensional matrix and an n-dimensional matrix, and C is an (m+n−2)-dimensional matrix.

Multidimensioinal matrix is the inner product of an m-dimensional matrix and an n-dimensional matrix.

Example 24.1:

$$\text{Suppose } A=[a_{ij}]_{32}=\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix},\ B=[b_j]_2=\begin{bmatrix} 7 \\ \\ 8 \end{bmatrix},$$

$$\text{then } A*_iB=\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} *_i \begin{bmatrix} 7 \\ \\ 8 \end{bmatrix} = \begin{bmatrix} 1*7+2*8 \\ 3*7+4*8 \\ 5*7+6*8 \end{bmatrix} = \begin{bmatrix} 23 \\ 53 \\ 83 \end{bmatrix}.$$

24.2.2 Middle products

Definition 24.2: Suppose $A=[a_{ij}]_{pq}$ is a 2-dimensional matrix, $B=[b_j]_q$ is a 1-dimensional matrix, A and B have one index, say *j*, in common, and both orders of *j* are q, then $C=[c_{ij}]_{pq}=[a_{ij}*b_j]_{pq}$, where $*$ is the ordinary multiplication, is called the middle product of A and B, denoted by $C=A*_mB$. C is a 2-dimensional matrix.

It is easy to generalize this definition to the middle product of an m-dimensional matrix and an n-dimensional matrix, and C is an (m+n−1)-dimensional matrix.

Example 24.2:

$$\text{Suppose } A=[a_{ij}]_{32}=\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix},\ B=[b_j]_2=\begin{bmatrix} 7 \\ \\ 8 \end{bmatrix},$$

$$\text{then } A*_mB=\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} *_m \begin{bmatrix} 7 \\ \\ 8 \end{bmatrix} = \begin{bmatrix} 1*7 & 2*8 \\ 3*7 & 4*8 \\ 5*7 & 6*8 \end{bmatrix} = \begin{bmatrix} 7 & 16 \\ 21 & 32 \\ 35 & 48 \end{bmatrix}.$$

24.2.3 Combined products

Definition 24.3: Suppose $A=[a_{ij}]_{22}$, $B=[b_{ij}]_{22}$, A and B have two indices *i* and *j* in

common, both orders of i are 2, both orders of j are also 2. Then A can be multiplied by B in three ways:

(1) For both common indices inner products are made: $C=A*_i^i*_i^jB=\sum_{i=1}^{2}\sum_{j=1}^{2}(a_{ij}*b_{ij})=(a_{11}b_{11}+a_{12}b_{12})+(a_{21}b_{21}+a_{22}b_{22})$. C is a 0-dimensional matrix.

(2) For both common indices middle products are made:

$$D=A*_m^i*_m^jB=\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}.$$ D is a 2-dimensional matrix.

(3) For some common index, say i, inner product is made, for the other common index, say j, middle product is made: $E=A*_i^i*_m^jB=[\sum_{i=1}^{2}(a_{ij}*b_{ij})]_2=[a_{11}b_{11}+a_{21}b_{21}, a_{12}b_{12}+a_{22}b_{22}]$. E is a 1-dimensional matrix.

The above three multiplications are called combined products of A and B, where $*_i^j$ denotes making inner product for the index j, $*_m^j$ denotes making middle product for the index j. The dimension of the combined product of A and B is: dimension 2 of A plus dimension 2 of B, then for every inner product, dimension 2 is subtracted, for every middle product, dimension 1 is subtracted, the remaining dimension is the dimension of the combined product of A and B.

It is easy to generalize the definition to the combined product of an m-dimensional matrix and an n-dimensional matrix.

Example 24.3:

Suppose $A=[a_{ij}]_{22}=\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B=[b_{ij}]_{22}=\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$. Then

$C=A*_i^i*_i^jB=\sum_{i=1}^{2}\sum_{j=1}^{2}(a_{ij}*b_{ij})=(a_{11}b_{11}+a_{12}b_{12})+(a_{21}b_{21}+a_{22}b_{22})=(1*5+2*6)=(3*7+4*8)=70.$

$$D=A*_m^i*_m^jB=\begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}=\begin{bmatrix} 1*5 & 2*6 \\ 3*7 & 4*8 \end{bmatrix}=\begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}.$$

$E=A*_i^i*_m^jB=[\sum_{i=1}^{2}(a_{ij}*b_{ij})]_2=[a_{11}b_{11}+a_{21}b_{21}, a_{12}b_{12}+a_{22}b_{22}]=[1*5+3*7, 2*6+4*8]=[26, 44].$

An m-dimensional matrix multiplied by an n-dimensional matrix, the resulting matrix is any dimension from $|m-n|$ to $m+n$.

24.2.4 Outer products

Definition 24.4:

Suppose $A=[a_i]_2^T=\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, where T denotes transpose, $B=[b_j]_2=[b_1, b_2]$, A and B have no

index in common, then $C=[a_i*b_j]_{22}=\begin{bmatrix} a_1b_1 & a_1b_2 \\ a_2b_1 & a_2b_2 \end{bmatrix}$, where $*$ is ordinary multiplication, is called the outer product of A and B, denoted by $A*_oB$. C is a 2-dimensional matrix.

It is easy to generalize the definition to the outer product of an m-dimensional matrix and an n-dimensional matrix.

Example 24.4:

Suppose $A=\begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $B=[3, 4]$, then $C=\begin{bmatrix} 1 \\ 2 \end{bmatrix}*_o[3, 4]=\begin{bmatrix} 1*3 & 1*4 \\ 2*3 & 2*4 \end{bmatrix}=\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$.

24.3 Isodimensional matrices

24.3.1 0-isodimensional matrices

0-isodimensional matrices refer to such matrices that a 0-dimensional matrix; i.e., a scalar can be multiplied only by a 0-dimensional matrix; i.e., a scalar, and yielding a 0-dimensional matrix; i.e., a scalar.

Example 24.5: Suppose a and b are two 0-dimensional matrices (scalars), then

$c=a*b$

is a 0-dimensional matrix (a scalar). This is an example of the outer product of scalars.

24.3.2 1-isodimensional matrices

1-isodimensional matrices refer to such matrices that a 1-dimensional matrix; i.e., a vector can be multiplied only by a 1-dimensional matrix; i.e., a vector, and yielding a 1-dimensional matrix; i.e., a vector.

Example 24.6: Suppose $V_1=[a_i]_n=[a_1, a_2, \ldots, a_n]$ and $V_2=[b_i]_n=[b_1, b_2, \ldots, b_n]$ are two 1-dimensional matrices; i.e., vectors, then

$V_3=V_1*_mV_2=[a_i]_n*_m[b_i]_n=[a_1, a_2, \ldots, a_n]*_m[b_1, b_2, \ldots, b_n]=[a_1b_1, a_2b_2, \ldots, a_nb_n]$

is a 1-dimensional matrix; i.e., a vector. This is an example of the middle product of vectors.

24.3.3 2-isodimensional matrices (planar matrices)

2-isodimensional matrices, also called planar matrices, refer to such matrices that a 2-dimensional matrix; i.e., a planar matrix can be multiplied only by a 2-dimensional matrix; i.e., a planar matrix, and yielding a 2-dimensional matrix; i.e., a planar matrix.

There are six multiplications for 2-isosimensional matrices: four inner products and two combined products. The four inner products are

$$C_1=[c_{ik}]_{pr}=[a_{ij}]_{pq}*_i[b_{jk}]_{qr}=[\sum_{j=1}^{q}(a_{ij}*b_{jk})]_{pr},\qquad(24.1)$$
$$C_2=[c_{ik}]_{pr}=[a_{ij}]_{pq}*_i[b_{kj}]_{rq}=[\sum_{j=1}^{q}(a_{ij}*b_{kj})]_{pr},$$
$$C_3=[c_{ik}]_{pr}=[a_{ji}]_{qp}*_i[b_{jk}]_{qr}=[\sum_{j=1}^{q}(a_{ij}*b_{jk})]_{pr},$$
$$C_4=[c_{ik}]_{pr}=[a_{ji}]_{qp}*_i[b_{kj}]_{rq}=[\sum_{j=1}^{q}(a_{ij}*b_{kj})]_{pr}.$$

Formula (24.1) is the most famous inner product studied by linear algebra. The two combined products are

$$C_5=[c_{ij}]_{pq}=[a_{ij}]_{pq}*_m^{i}*_m^{j}[b_{ij}]_{pq}=[a_{ij}*b_{ij}]_{pq},$$
$$C_6=[c_{ij}]_{pq}=[a_{ij}]_{pq}*_m^{i}*_m^{j}[b_{ji}]_{qp}=[a_{ij}*b_{ji}]_{pq}.$$

24.3.4 3-isodimensional matrices (cuboidal matrices)

3-isodimensional matrices, also called cuboidal matrices, refer to such matrices that a 3-dimensional matrix; i.e., a cuboidal matrix can be multiplied only by a 3-dimensional matrix; i.e., a cuboidal matrix, and yielding a 3-dimensional matrix; i.e., a cuboidal matrix.

24.3.4.1 Addition of cuboidal matrices

Definition 24.5: Suppose $A=[a_{ijk}]_{pqr}$ and $B=[b_{ijk}]_{pqr}$ are two cuboidal matrices, then $C=[c_{ijk}]_{pqr}=[a_{ijk}+b_{ijk}]_{pqr}$ is called the sum of A and B, denoted by C=A+B.

The addition of cuboidal matrices satisfies

Associative law: A+B+C=A+(B+C).

Commutative law: A+B=B+A.

If all of its elements are zero, then the cuboidal matrix is called the cuboidal zero matrix, denoted by 0_{pqr} or 0. Obviously, for any cuboidal matrix A, A+0=0+A holds.

$$\begin{bmatrix} -a_{111} & -a_{121} & \cdots & -a_{1q1} \\ -a_{211} & -a_{221} & \cdots & -a_{2q1} \\ \cdots & \cdots & \cdots & \cdots \\ -a_{p11} & -a_{p21} & \cdots & -a_{pq1} \end{bmatrix}$$
$$\vdots$$
$$\begin{bmatrix} -a_{11r} & -a_{12r} & \cdots & -a_{1qr} \\ -a_{21r} & -a_{22r} & \cdots & -a_{2qr} \\ \cdots & \cdots & \cdots & \cdots \\ -a_{p1r} & -a_{p2r} & \cdots & -a_{pqr} \end{bmatrix}$$

is called the minus cuboidal matrix of A, denoted by −A. Obviously, A+(−A)=0 holds. The subtraction of cuboidal matrices is defined as A−B=A+(−B).

24.3.4.2 Multiplications of cuboidal matrices

There are two types of combined products for cuboidal matrices. The first type is that

cuboidal matrices A and B have three indices, say, *i*, *j*, and *k* in common, and for all of them middle products are made. The second type is that A and B have two indices, say, *j* and *k* in common, and for one of them inner product is made, for the other middle product is made. Here, we discuss only two multiplications of the second type that are more useful.

Definition 24.6: Suppose $A=[a_{ijk}]_{pqr}$, $B=[b_{jkl}]_{qrs}$, then $C=[c_{ijl}]_{pqs}=A*_m^j*_i^k B=[\sum_{k=1}^{r}(a_{ijk}*b_{jkl})]_{pqs}$, where

$$c_{ijl}=\sum_{k=1}^{r}(a_{ijk}*b_{jkl}) \quad (i=1, 2, \ldots, p;\ j=1, 2, \ldots, q;\ l=1, 2, \ldots, s), \tag{24.2}$$

in which $*$ is the ordinary multiplication, is called the product of A right-contracted multiplied by B, denoted by $C=A*_{rc}B$.

Definition 24.7: Suppose $A=[a_{ijk}]_{pqr}$, $B=[b_{jkl}]_{qrs}$, then $D=[d_{ikl}]_{prs}=A*_i^j*_m^k B=[\sum_{j=1}^{q}(a_{ijk}*b_{jkl})]_{prs}$, where

$$d_{ikl}=\sum_{j=1}^{q}(a_{ijk}*b_{jkl})(i=1, 2, \ldots, p;\ k=1, 2, \ldots, r;\ l=1, 2, \ldots, s), \tag{24.3}$$

in which $*$ is the ordinary multiplication, is called the product of A left-contracted multiplied by B, denoted by $D=A*_{lc}B$.

Both $*_{rc}$ and $*_{lc}$ require that the width of A equals the height of B and the depth of A equals the width of B. Then we say that A is left multipliable with B and B is right multipliable with A or that A and B are multipliable. The relationships among the height, width, and depth of A, B, and C in $C=A*_{rc}B$ are shown in Fig. 24.9.

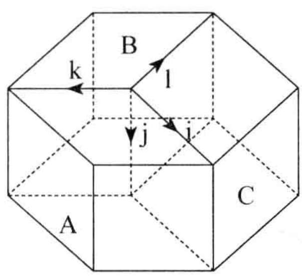


Fig. 24.9 Relationships among the height, width, and depth of A, B, and C in $C=A*_{rc}B$

Figs. 24. 10 and 24.11 show $C_2=A_2*_{rc}B_2$ and $D_2=A_2*_{lc}B_2$ respectively, where A_2, B_2, C_2, and D_2 stand for that they are cubic matrices of order 2.

$*_{lc}$ and $*_{rc}$ satisfy neither commutative law nor associative law, but they satisfy left-right associative law:

$$(A*_{lc}B)*_{rc}C=A*_{lc}(B*_{rc}C) \tag{24.4}$$

Prove (24.4). Proof: Let $A=[a_{ijk}]_{pqr}$, $B=[b_{jkl}]_{qrs}$, $C=[c_{klm}]_{rst}$. Let $E=A*_{lc}B=[e_{ikl}]_{prs}$, $F=B*_{rc}C=[f_{jkm}]_{qrt}$, where

$$e_{ikl}=\sum_{j=1}^{q}(a_{ijk}b_{jkl}) \quad (i=1, 2, \ldots, p;\ k=1, 2, \ldots, r;\ l=1,.\ 2, \ldots, s),$$

$$f_{jkm}=\sum_{l=1}^{s}(b_{jkl}c_{klm}) \quad (j=1, 2, \ldots, q;\ k=1, 2, \ldots, r;\ m=1, 2, \ldots, t).$$

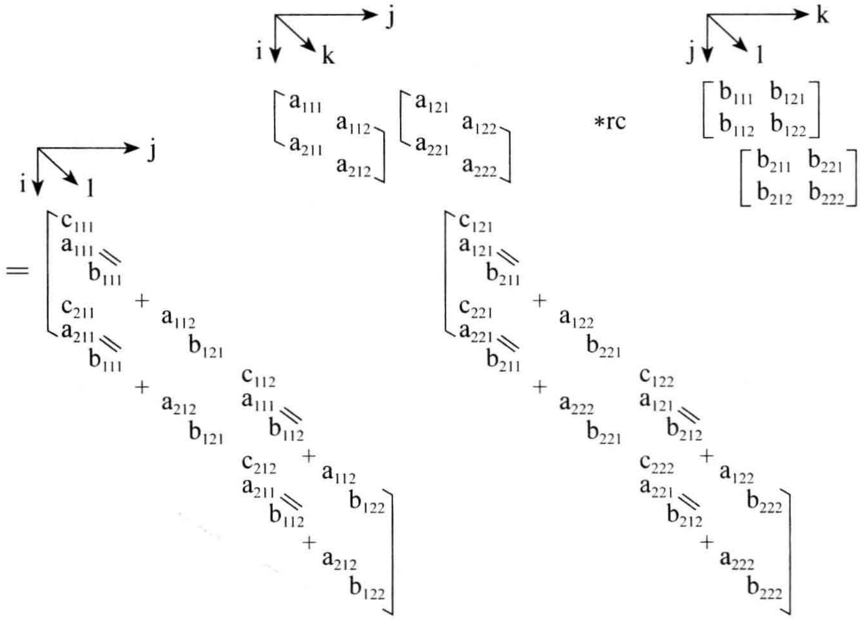


Fig. 24.10 $C_2=A_2*_{rc}B_2$

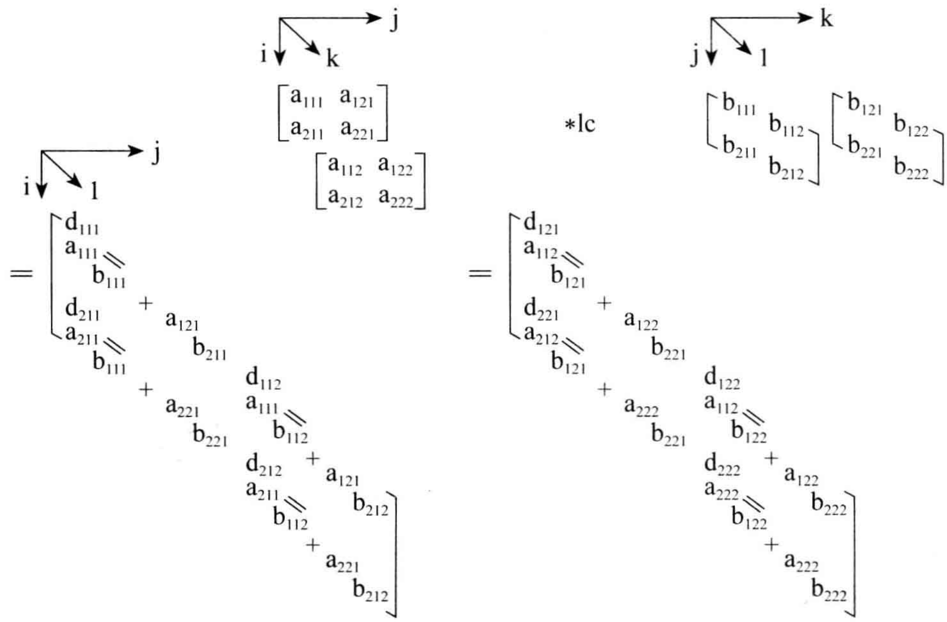


Fig. 24.11 $D_2=A_2*_{lc}B_2$

Thus, the *i*th height, *k*th width, and *m*th depth element in $(A*_{lc}B)*_{rc}C=E*_{rc}C$ is

$$\sum_{l=1}^{s}(e_{ikl}c_{klm})=\sum_{l=1}^{s}\sum_{j=1}^{q}(a_{ijk}b_{jkl}c_{klm}).$$

And the *i*th height, *k*th width, and *m*th depth element in $A*_{lc}(B*_{rc}C)=A*_{lc}F$ is

$$\sum_{j=1}^{q}(a_{ijk}f_{jkm})=\sum_{j=1}^{q}\sum_{l=1}^{s}(a_{ijk}b_{jkl}c_{klm}).$$

The two additive signs can change order, hence

$\sum_{l=1}^{s}\sum_{j=1}^{q}(a_{ijk}b_{jkl}c_{klm})=\sum_{j=1}^{q}\sum_{l=1}^{s}(a_{ijk}b_{jkl}c_{klm})$.

And i, k, and m are arbitrary, therefore (24.9) is proved.

Q.E.D.

Unlike what we do in planar matrix, we cannot define powers of cuboidal matrices, because none of $*_{lc}$ and $*_{rc}$ satisfies associative law.

Both $*_{lc}$ and $*_{rc}$ satisfy left and right distributive law with respect to +:

$A*_{lc}(B+C)=A*_{lc}B+A*_{lc}C$

$A*_{rc}(B+C)=A*_{rc}B+A*_{rc}C$

$(B+C)*_{lc}A=B*_{lc}A+C*_{lc}A$

$(B+C)*_{rc}A=B*_{rc}A+C*_{rc}A$.

24.3.4.3 Cuboidal unity matrices

A cuboidal unity matrix is such a matrix that certain elements in it are 1s, others are 0s. A p*q*r cuboidal matrix with all elements in it are 1s is called a cuboidal all-one unity matrix, denoted by U_{pqr} or U. Cuboidal zero matrix 0_{pqr} is another special kind of cuboidal unity matrix.

Definition 24.8: Suppose $A=[a_{ijk}]_{pqq}$, then the plane made of elements a_{ijj} (i=1, 2, …, p; j=1, 2, …, q), the main diagonal lines of every height plane A_{Hi} (i=1, 2, …, p) of A, is called the main diagonal plane with respect to height.

Similarly, we can define the main diagonal plane with respect to width and the main diagonal plane with respect to depth.

Definition 24.9: A cuboidal matrix is called cuboidal unity matrix of the main diagonal plane with respect to height, denoted by E_{Hpqq}, if all of its elements on the main diagonal plane with respect to height are 1s and all of the other elements are 0s. If the cuboidal unity matrix of the main diagonal plane with respect to height is a cubic matrix, then it is denoted by E_H.

Similarly, we can define cuboidal unity matrix of the main diagonal plane with respect to width E_{Wpqp} and E_W, cuboidal unity matrix of the main diagonal plane with respect to depth E_{Dppr} and E_D.

Definition 24.10: A cubic matrix of order p is called the cubic unity matrix of the main diagonal axis, denoted by I_p or I, if all of its elements a_{iii} (i=1, 2, …, p) of the main diagonal axis are 1s and all of the other elements are 0s.

Fig. 24.12 shows E_H, E_W, E_D, I_2, U_2, and 0_2 of order 2.

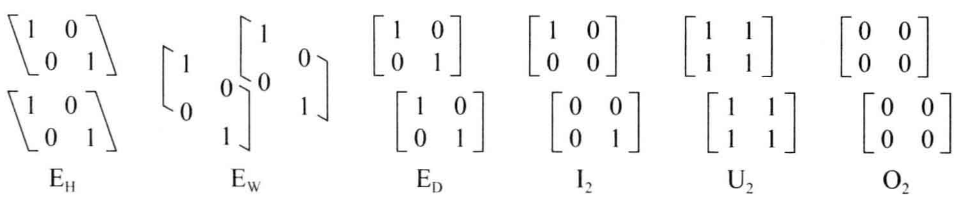


Fig. 24.12 E_H, E_W, E_D, I_2, U_2, and 0_2 of order 2

The operations of $*lc$ and $*rc$ on the six cubic unity matrices of order p are shown in Tables 24.1 and 24.2 where the matrices in the form of pA are scalar cubic matrices which will be discussed later.

Table 24.1 $*_{IC}$ **on unity matrices**

$*_{lc}$	0	I	E_H	E_D	E_W	U
0	0	0	0	0	0	0
I	0	I	I	E_D	I	E_D
E_H	0	E_H	E_H	U	E_H	U
E_D	0	I	E_H	E_D	E_W	U
E_W	0	I	pI	E_D	E_D	pE_D
U	0	E_H	pE_H	U	U	pU

Table 24.2 $*_{TC}$ **on unity matrices**

$*_{rc}$	0	I	E_H	E_D	E_W	U
0	0	0	0	0	0	0
I	0	I	I	E_D	I	E_D
E_H	0	E_H	E_H	U	E_H	U
E_D	0	I	E_D	E_D	pI	pE_D
E_W	0	I	E_W	E_D	E_H	U
U	0	E_H	U	U	pE_H	U

Definition 24.11: A cuboidal matrix A satisfying

$$A*_{lc}A=A \tag{24.5}$$

is called an idempotent cuboidal matrix with respect to $*_{lc}$. A cuboidal matrix satisfying

$$A*_{rc}A=A \tag{24.6}$$

is called an idempotent cuboidal matrix with respect to $*_{rc}$. If both (24.5) and (24.6) are satisfied, then A is called an idempotent cuboidal matrix which must be a cubic matrix.

From Tables 24.1 and 24.2 we know that 0, I, E_H, and E_D are all idempotent cubic matrices, called idempotent cubic unity matrices. While U is an idempotent cubic unity matrix with respect to $*_{rc}$.

24.3.4.4 Scalar products and scalar cubic matrices

Definition 24.12: The cuboidal matrix

$$\begin{bmatrix} ga_{111} & ga_{121} & \cdots & ga_{1q1} \\ ga_{211} & ga_{221} & \cdots & ga_{2q1} \\ \cdots & \cdots & \cdots & \cdots \\ ga_{p11} & ga_{p21} & \cdots & ga_{pq1} \end{bmatrix}$$

$$\cdot$$

$$\cdot$$

$$\begin{bmatrix} ga_{11r} & ga_{12r} & \cdots & ga_{1qr} \\ ga_{21r} & ga_{22r} & \cdots & ga_{2qr} \\ \cdots & \cdots & \cdots & \cdots \\ ga_{p1r} & ga_{p2r} & \cdots & ga_{pqr} \end{bmatrix}$$

 is called the scalar product of number g and cuboidal matrix $A=[a_{ijk}]$, denoted by gA. The

scalar product gA is the outer product of a 0-dimensional matrix g and a 3-dimensional matrix A. Scalar products satisfy the following laws:

$(g_1+g_2)A=g_1A+g_2A$

$g(A+B)=gA+gB$

$g_1(g_2A)=(g_1g_2)A$

$1A=A$

$g(A*_{lc}B)=(gA)*_{lc}B=A*_{lc}(gB)$

$$g(A*_{rc}B)=(gA)*_{rc}B=A*_{rc}(gB) \qquad (24.7)$$

Prove (24.7): Proof: Let $A=[a_{ijk}]_{pqr}$, $B=[b_{jkl}]_{qrs}$. The element of the *i*th height, *j*th width, and *l*th depth in $g(A*_{rc}B)$, $(gA)*_{rc}B$, and $A*_{rc}(gB)$ are

$g\sum_{k=1}^{r}(a_{ijk}b_{jkl})$

$\sum_{k=1}^{r}[(ga_{ijk})b_{jkl}]$

$\sum_{k=1}^{r}[a_{ijk}(gb_{jkl})]$

respectively. Obviously, they are equal.

Q.E.D.

Definition 24.13: The scalar product of a number g and a cubic unity matrix Λ is called scalar cubic matrix, denoted by $g\Lambda$.

There are six scalar cubic matrices: $g0$, gI, gE_H, gE_D, gE_W, gU. Scalar cubic matrices satisfy:

$g_1\Lambda+g_2\Lambda=(g_1+g_2)\Lambda$.

That is, the addition of scalar cubic matrices can be reduced to the addition of numbers.

The scalar cubic matrices $g0$, gI, gE_H, and gE_D are the scalar products of number g and idempotent cubic unity matrices, satisfying the following laws:

$(g_10)*_{lc}(g_20)=(g_1g_2)(0*_{lc}0)=(g_1g_2)0$

$(g_10)*_{rc}(g_20)=(g_1g_2)(0*_{rc}0)=(g_1g_2)0$

$(g_1I)*_{lc}(g_2I)=(g_1g_2)(I*_{lc}I)=(g_1g_2)I$

$(g_1I)*_{rc}(g_2I)=(g_1g_2)(I*_{rc}I)=(g_1g_2)I$

$(g_1E_H)*_{lc}(g_2E_H)=(g_1g_2)(E_H*_{lc}E_H)=(g_1g_2)E_H$

$(g_1E_H)*_{rc}(g_2E_H)=(g_1g_2)(E_H*_{rc}E_H)=(g_1g_2)E_H$

$(g_1E_D)*_{lc}(g_2E_D)=(g_1g_2)(E_D*_{lc}E_D)=(g_1g_2)E_D$

$(g_1E_D)*_{rc}(g_2E_D)=(g_1g_2)(E_D*_{rc}E_D)=(g_1g_2)E_D$.

That is, the multiplications of scalar cubic matrices such as $g0$, gI, gE_H, and gE_D can be reduced to the multiplications of numbers. In addition, for gU,

$(g_1U)*_{rc}(g_2U)=(g_1g_2)(U*_{rc}U)=(g_1g_2)U$

holds.

24.3.4.5 Transposes of cuboidal matrices and transpose group

In planar matrix theory, the planar matrix obtained by interchanging rows and columns of a planar matrix A is called the transpose of A. Here, we call it transpose of planar matrix in order to distinguish it from the plane transpose of cuboidal matrix.

Definition 24.14: Suppose cuboidal matrix $A=[a_{ijk}]_{pqr}=$

$$\begin{array}{c}
\begin{bmatrix} a_{111} & a_{121} & \cdots & a_{1q1} \\ a_{112} & a_{122} & \cdots & a_{1q2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{11r} & a_{12r} & \cdots & a_{1qr} \end{bmatrix} \\
\begin{bmatrix} a_{211} & a_{221} & \cdots & a_{2q1} \\ a_{212} & a_{222} & \cdots & a_{2q2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{21r} & a_{22r} & \cdots & a_{2qr} \end{bmatrix} \\
\vdots \\
\begin{bmatrix} a_{p11} & a_{p21} & \cdots & a_{pq1} \\ a_{p12} & a_{p22} & \cdots & a_{pq2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p1r} & a_{p2r} & \cdots & a_{pqr} \end{bmatrix}
\end{array}$$

The cuboidal matrix $[a_{ikj}]_{prq}=$

$$\begin{array}{c}
\begin{bmatrix} a_{111} & a_{112} & \cdots & a_{11r} \\ a_{121} & a_{122} & \cdots & a_{12r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1q1} & a_{1q2} & \cdots & a_{1qr} \end{bmatrix} \\
\begin{bmatrix} a_{211} & a_{212} & \cdots & a_{21r} \\ a_{221} & a_{222} & \cdots & a_{22r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{2q1} & a_{2q2} & \cdots & a_{2qr} \end{bmatrix} \\
\vdots \\
\begin{bmatrix} a_{p11} & a_{p12} & \cdots & a_{p1r} \\ a_{p21} & a_{p22} & \cdots & a_{p2r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{pq1} & a_{pq2} & \cdots & a_{pqr} \end{bmatrix}
\end{array}$$

obtained by making transposes of planar matrix to every height plain A_{Hi} (i=1, 2, …, p) of A is called the height plane transpose of cuboidal matrix A, denoted by A^{Th}.

Similarly, we can define the width plane transpose of cuboidal matrix A, denoted by A^{Tw} and the depth plane transpose of cuboidal matrix A, denoted by A^{Td}. T_h, T_w, and T_d are called by a joint name plane transpose of cuboidal matrix or shape-deforming transpose, denoted by T_p.

A^{Th} can be viewed as being obtained from A by interchanging the width and depth of A. A^{Tw} can be viewed as being obtained from A by interchanging the height and depth of A. A^{Td} can be viewed as being obtained from A by interchanging the height and width of A.

T_h is such a transpose that keeps it fixed the positions of the elements on the main diagonal plane with respect to height of A_{pqq}. T_w is such a transpose that keeps it fixed the positions of the elements on the main diagonal plane with respect to width of A_{pqp}. T_d is such a transpose that keeps it fixed the positions of the elements on the main diagonal plane with respect to depth of A_{ppr}.

Definition 24.15: Taking the line connecting the elements a_{111} and a_{pqr} in the cuboidal matrix $A=[a_{ijk}]_{pqr}$ mentioned in Definition 24.14 as an axis called the main diagonal axis, the cuboidal matrix $[a_{kij}]_{rpq}$

$$\left[\begin{array}{cccc} a_{111} & & & \\ a_{112} & a_{121} & & \\ \cdot & a_{122} & \cdot & \\ a_{11r} & \cdot & \cdot & a_{1q1} \\ & a_{12r} & \cdot & a_{2q2} \\ & & \cdot & \cdot \\ & & & a_{1qr} \end{array}\right] \left[\begin{array}{cccc} a_{211} & & & \\ a_{212} & a_{221} & & \\ \cdot & a_{222} & \cdot & \\ a_{21r} & \cdot & \cdot & a_{2q1} \\ & a_{22r} & \cdot & a_{2q2} \\ & & \cdot & \cdot \\ & & & a_{2qr} \end{array}\right] \cdots\cdots \left[\begin{array}{cccc} a_{p11} & & & \\ a_{p12} & a_{p21} & & \\ \cdot & a_{p22} & \cdot & \\ a_{p1r} & \cdot & \cdot & a_{pq1} \\ & a_{p2r} & \cdot & a_{pq2} \\ & & \cdot & \cdot \\ & & & a_{pqr} \end{array}\right]$$

obtained by rotating A 120° around the axis clockwise viewed from a_{111} to a_{pqr} is called the clockwise rotation transpose of A or the right rotation transpose of A, denoted by A^{Tr}. The cuboidal matrix $[a_{jki}]_{qrp}$

$$\begin{bmatrix} a_{111} & a_{112} & \cdots & a_{qqr} \\ a_{121} & a_{122} & \cdots & a_{12r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1q1} & a_{1q2} & \cdots & a_{1q4} \end{bmatrix}$$

$$\begin{bmatrix} a_{211} & a_{212} & \cdots & a_{21r} \\ a_{221} & a_{222} & \cdots & a_{22r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{2q1} & a_{2q2} & \cdots & a_{2qr} \end{bmatrix}$$

$$\vdots$$

$$\begin{bmatrix} a_{p11} & a_{p12} & \cdots & a_{p1r} \\ a_{p21} & a_{p22} & \cdots & a_{p2r} \\ \cdots & \cdots & \cdots & \cdots \\ a_{pq1} & a_{pq2} & \cdots & a_{pqr} \end{bmatrix}$$

obtained by rotating A 120° around the axis anticlockwise viewed from a_{111} to a_{pqr} is called the anticlockwise rotation transpose of A or the left rotation transpose of A, denoted by A^{Tl}.

A^{Tr} can be viewed as being obtained from A by rotating rightward the three subscripts of a_{ijk}. A^{Tl} can be viewed as being obtained from A by rotating leftward the three subscripts of a_{ijk}. T_r and T_l are called by a joint name axial transpose of cuboidal matrix, denoted by T_a. T_a is such a transpose that keeps it fixed the positions of the elements on the main diagonal axis of the cubic matrix A_p.

Definition 24.16: The transpose that keeps it fixed the positions of all the elements in A is called the stationary transpose, denoted by A^{T0}.

Obviously, we have

$A^{T0}=A$.

T_0 and T_a are called by a joint name cuboid transpose of cuboidal matrix or shape-preserving transpose, denoted by T_c. T_c and T_p are called by a joint name transpose of cuboidal matrix, denoted by T.

The classification of transposes of cuboidal matrix is shown in Fig. 24.13.

$$T\begin{cases} T_c\begin{cases} T_o \\ T_a\begin{cases} T_l \\ T_r \end{cases} \end{cases} \\ T_p\begin{cases} T_h \\ T_w \\ T_d \end{cases} \end{cases}$$

Fig. 24.13 classification of transposes of cuboidal matrix

Transpose T can be regarded as a unary operation on cuboidal matrices. Transpose T on cubic unity matrices is shown in Table 24.3.

Prove $E_H^{Tr}=E_W$.

Proof: $E_H^{Tr}=[e_{ijj}]_{pqq}^{Tr}=[e_{jij}]_{qpq}=E_W$.

Q.E.D.

The other transposes in Table 24.3 can be proved similarly.

Table 24.3 Transpose T on cubic unity matrices

	T_o	T_h	T_w	T_d	T_r	T_l
0	0	0	0	0	0	0
I	I	I	I	I	I	I
E_H	E_H	E_H	E_D	E_W	E_W	E_D
E_W	E_W	E_D	E_W	E_H	E_D	E_H
E_D	E_D	E_W	E_H	E_D	E_H	E_W
U	U	U	U	U	U	U

Definition 24.17: Transposing the transposed matrix of cuboidal matrix A, obtain the composite transpose of A.

Composition of transposes of cuboidal matrices "o" can be regarded as a binary operation on the set of transposes of cuboidal matrices S_T, its operation table is shown in Table 24.4.

Table 24.4 ⟨S_T, o⟩

o	T_o	T_d	T_w	T_h	T_r	T_l
T_o	T_o	T_d	T_w	T_h	T_r	T_l
T_d	T_d	T_o	T_r	T_l	T_w	T_h
T_w	T_w	T_l	T_o	T_r	T_h	T_d
T_h	T_h	T_r	T_l	T_o	T_d	T_w
T_r	T_r	T_h	T_d	T_w	T_l	T_o
T_l	T_l	T_w	T_h	T_d	T_o	T_r

Table 24.5 ⟨S_T, ◇⟩

◇	p_1	p_2	p_3	p_4	p_5	p_6
p_1	p_1	p_2	p_3	p_4	p_5	p_6
p_2	p_2	p_1	p_5	p_6	p_3	p_4
p_3	p_3	p_6	p_1	p_5	p_4	p_2
p_4	p_4	p_5	p_6	p_1	p_2	p_3
p_5	p_5	p_4	p_2	p_3	p_6	p_1
p_6	p_6	p_3	p_4	p_2	p_1	p_5

Prove $T_h o T_l = T_w$.

Proof: $([a_{ijk}]^{Th})^{Tl}=[a_{ikj}]^{Tl}=[a_{kji}]$. And $[a_{ijk}]^{Tw}=[a_{kji}]$. Hence, $([a_{ijk}]^{Th})^{Tl}=[a_{ijk}]^{Tw}$. I, j,and k are arbitrary, therefore $(A^{Th})^{Tl}=A^{Tw}$, that is, $T_h o T_l = T_w$.

Q.E.D.

Theorem 24.1: ⟨S_T, o⟩ is a group.

Proof: First, ⟨S_3, ◇⟩ is a symmetric group of degree 3 whose operation table is shown in Table 24.5 and ⟨S_T, o⟩ is an algebra because o is closed with respect to S_T, shown in Table 24.4.

Secondly, there is a bijection f: $S_3 \rightarrow S_T$:

$f(p_1)=T_o$, $f(p_2)=T_d$, $f(p_3)=T_w$, $f(p_4)=T_h$, $f(p_5)=T_r$, $f(p_6)=T_l$.

Thirdly, the operations preserve under f; i.e.,

$f(p_i \diamond p_j)=f(p_i) \text{ o } f(p_j)$

holds for i, j=1, 2, …, 6. For example,

$$f(p_2 \Diamond p_4)=f(p_6)=T_l$$

and

$$f(p_2) \text{ o } f(p_4)=T_d \text{o} T_h=T_l.$$

Hence,

$$f(p_2 \Diamond p_4)= f(p_2) \text{ o } f(p_4).$$

Fourthly, the algebraic constant p_1 in S_3 maps into the algebraic constant T_o in S_T.

Therefore, f is an isomorphism from ⟨S_3, ◇⟩ to ⟨S_T, o⟩. Hence, ⟨S_T, o⟩ is a group.

Q.E.D.

The cuboid transposes of cuboidal matrices form a set S_c={T_o, T_r, T_l}. ⟨S_c, o⟩ is an invariant subgroup of ⟨S_T, o⟩. Sc divides S_T into two cosets: T_oS_c={T_o, T_r, T_l} and T_wS_c={T_w, T_h, T_d}. Let T_o and T_w, T_r and T_h, T_l and T_d be cotransposes mutually, then we have the following formulae of transposes of multiplications:

$$(A*_{lc}B)^T=(B^{Tl}*_{lc}A^{Tr})^{Tco} \tag{24.8}$$

$$(A*_{rc}B)^T=(B^{Tl}*_{rc}A^{Tr})^{Tco} \tag{24.9}$$

$$(A*_{lc}B)^T=(B^{Tw}*_{rc}A^{Tw})^{Tco} \tag{24.10}$$

$$(A*_{rc}B)^T=(B^{Tw}*_{lc}A^{Tw})^{Tco} \tag{24.11}$$

Where T is a transpose, T_{co} is its cotranspose. Let T=T_w, T_{co}=T_o, then formulae (24.8) through (24.11) become:

$$(A*_{lc}B)^{Tw}=B^{Tl}*_{lc}A^{Tr} \tag{24.12}$$

$$(A*_{rc}B)^{Tw}=B^{Tl}*_{rc}A^{Tr} \tag{24.13}$$

$$(A*_{lc}B)^{Tw}=B^{Tw}*_{rc}A^{Tw} \tag{24.14}$$

$$(A*_{rc}B)^{Tw}=B^{Tw}*_{lc}A^{Tw} \tag{24.15}$$

respectively. Formulae (24.12) through (24.15) show that transposes of multiplications equal multiplications of transposes.

Prove (24.13).

Proof: The ith height, jth width, and lth depth element in $A*_{rc}B$ is

$$\textstyle\sum_{k=1}^{r}(a_{ijk}b_{jkl}),$$

therefore, the ith height, jth width, and lth depth element in$(A*_{rc}B)^{Tw}$ is

$$\textstyle\sum_{k=1}^{r}(a_{ljk}b_{jki}).$$

On the other hand, the ith height, jth width, and kth depth element in B^{Tl} is b_{jki}, the jth height, kth width, and lth depth element in A^{Tr} is a_{ljk}, therefore, the ith height, jth width, and lth depth element in $B^{Tl}*_{rc}A^{Tr}$ is

$$\textstyle\sum_{k=1}^{r}(a_{jki}b_{ljk})=\sum_{k=1}^{r}(a_{ljk}b_{jki}).$$

Q.E.D.

Transposes of addition of cuboidal matrices satisfy:

$$(A+B)^T=A^T+B^T.$$

Transposes of scalar cubic matrices satisfy:

$(g\Lambda)^T = g(\Lambda)^T$.

24.3.4.6 Cubic identity matrices

Definition 24.18: Suppose

$E*'A=A^T$

where E is a cubic matrix, A is right multipliable with E, then E is called a left identity, denoted by E_l. If T is T_o, T_h, T_w, T_d, T_r, and T_l respectively, then E is E_{To}, E_{Th}, E_{Tw}, E_{Td}, E_{Tr}, and E_{Tl}, respectively. If $*'$ is $*_{lc}$, then E is E_{*lc}, otherwise, E is E_{*rc}.

Example 24.7: Suppose

$E*_{lc}A=A^{To}$

then E is denoted by $E_{*lc,To,l}$.

Definition 24.19: Suppose

$A*'E=A^T$

where E is a cubic matrix, A is left multipliable with E, then E is called a right identity, denoted by E_r. If T is T_o, T_h, T_w, T_d, T_r, and T_l respectively, then E is E_{To}, E_{Th}, E_{Tw}, E_{Td}, E_{Tr}, and E_{Tl}, respectively. If $*'$ is $*_{lc}$, then E is E_{*lc}, otherwise, E is E_{*rc}.

Definition 24.20: Suppose

$$\begin{cases} E*'A=A^{T'} \\ A*''E=A^{T''} \end{cases}$$

Then E is called a two-sided identity, denoted by E_{lr}.

Theorem 24.2 (identity existence theorem): There exist four identities: $E_{*lc,To,l}=E_D$, $E_{*lc,Th,r}=E_{*rc,Td,l}=E_W$, $E_{*rc,To,r}=E_H$.

Proof: For any element a_{ijk} in $A=[a_{ijk}]_{pqr}$, an axial transpose is such that changes all of its three subscripts, either by rotating left or rotating right; a plane transpose is such that interchanges two subscripts and leaves the third unchanged; a stationary transpose is such that keeps all of its three subscripts unchanged.

Now, we prove that for E_{*rc} and E_r, there exists only E_{To}; ie., $E_{*rc,r,To}=E_H$.

An element c_{ijl} in $C=A*_{rc}B$ is determined by (24.2):

$$c_{ijl}=\sum_{k=1}^{r}(a_{ijk}b_{jkl}) \quad (i=1, 2, \ldots, p; j=1, 2, \ldots, q; l=1, 2, \ldots, s). \tag{24.2}$$

If B wants to be E_r, then it must satisfy:

$C=A*_{rc}B=A^T$.

In (24.2), the first two subscripts of c_{ijl} are the same as those of a_{ijk}, therefore it is not possible to obtain C from A by making axial transpose or plane transpose on A, that is, it is not possible for B to be an identity other than E_{To}. If B wants to be E_{To}, then it must satisfy:

$c_{ijl}=\sum_{k=1}^{r}(a_{ijk}b_{jkl})=a_{ijl}$.

From Definition 24.19, we learn that B is a cubic matrix, that is. We have q=r=s. This is

to say that j, k, and l are all from 1 to r. Thus, for any l there exists $k=l$ so that the three subscripts of a_{ijk} are the same as those of c_{ijl}. Let

$$b_{jkl}=\begin{cases} 0, & k\neq l \\ 1, & k=l. \end{cases} \qquad (24.16)$$

Then (24.2) becomes:

$$c_{ijl}=a_{ij1}*0+a_{ij2}*0+\ldots+a_{ijl}*1+\ldots+a_{ijr}*0=a_{ijl}.$$

And c_{ijl} is an arbitrary element in C. Therefore, B determined by (24.16) is indeed $E_{*rc,To,r}$. And from Table 24.2, we learn that B is just E_H. Therefore, $E_{*rc,To,r}=E_H$ and it is the only E_r for E_{*rc}.

Similarly, we can prove $E_{*rc,Td,l}=E_W$, $E_{*lc,To,l}=E_D$, and $E_{*lc,Th,r}=E_W$.

Q.E.D.

From Theorem 24.2 we learn that E_W is a a two-sided identity, satisfying:

$$\begin{cases} E_W *_{rc} A=A^{Td} \\ A *_{lc} E_W=A^{Th}. \end{cases}$$

Example 24.8: E_H is $E_{*rc,To,r}$ shown in Fig. 24.14.

$$\begin{bmatrix} a_{111} & a_{112} \\ a_{211} & a_{212} \end{bmatrix} \begin{bmatrix} a_{121} & a_{122} \\ a_{221} & a_{222} \end{bmatrix} *rc \begin{matrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{matrix} = \begin{bmatrix} a_{111} & a_{112} \\ a_{211} & a_{212} \end{bmatrix} \begin{bmatrix} a_{121} & a_{122} \\ a_{221} & a_{222} \end{bmatrix}$$

Fig. 24.14 E_H is $E_{*rc,To,r}$

24.3.4.7 Inverse cubic matrices and cuboidal matrix equations

Definition 24.21: If every height plane A_{Hi} (i=1, 2, …, p) of the cuboidal matrix A is a full rank planar matrix, then A is called full rank cuboidal matrix with respect to height.

Similarly, we can define full rank cuboidal matrix with respect to width and full rank cuboidal matrix with respect to depth.

Like in planar matrix theory, in cuboidal matrix theory, the introduction of inverse cubic matrix is also to solve cuboidal matrix equations in the form of

$A*'X=C$

or

$Y*'A=D$.

Definition 24.22: Suppose there is a cuboidal matrix equation:

$$A*'X=C \qquad (24.17)$$

where A is a cubic matrix of order p, X and C are p*p*t matrices. If there exists an order p cubic matrix B such that B left multiplies the two sides of (24.17), obtaining:

$B*''(A*'X)=E*'X=X^{T}=B*''C$,

then B is called the left inverse of A with respect to $*''*'$ and E, denoted by $B=A^{-1}_{*''*',E,l}$. If $*''$ is associative with respect to $*'$, then B is called an associative inverse, otherwise, B is a nonassociative inverse.

Definition 24.23: Suppose there is a cuboidal matrix equation;

$$Y*'A=C \tag{24.18}$$

where A is an order q matrix, Y and C are $p*q*q$ cuboidal matrices. If there exists an order q cubic matrix B such that B right $*''$ multiplies the two sides of (24.18), obtaining:

$$(Y*'A)*''B=Y*'E=Y^T=C*''B,$$

then B is called the right inverse of A with respect to $*''*'$ and E, denoted by $B=A^{-1}_{*''*',E,r}$. If $*''$ is associative with $*'$, then B is called an associative inverse, otherwise, it is a nonassociative inverse.

Theorem 24.3: Suppose $A=[a_{jkl}]_{ppp}$, $X=[x_{klm}]_{ppt}$, $C=[c_{jlm}]_{ppt}$, then if and only if A is full rank with respect to depth, then A in

$$A*_{lc}X=C \tag{24.19}$$

has a nonassociative inverse:

$$B=[b_{ijl}]_{ppp}=A^{-1}_{*lc*lc,\ ED,\ l}$$

where

$$b_{ijl}=\frac{\Delta b_{ijl}}{\Delta A^T_{Dl}} \tag{24.20}$$

$$\Delta b_{ijl}=\Delta A^T_{Dl}\langle j\rangle \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix} \begin{matrix} \langle k=i\rangle \\ \\ \\ \\ \\ \\ \\ \end{matrix} \tag{24.21}$$

(As to the meaning of (24.21), see the proof of Theorem 24.3).

Proof: From Definition 24.22 we know that finding the left inverse of A is equivalent to solving equation (24.17). Now, we solve (24.19).

B left $*lc$ multiplies the two sides of (24.19), obtaining:

$$B*_{lc}(A*_{lc}X)=B*_{lc}C. \tag{24.22}$$

Let the $p*p*t$ matrix

$$Z=A*_{lc}X$$

where

$$z_{jlm}=\sum_{k=1}^{p}(a_{jkl}x_{klm}) \quad (j,\ l=1,\ 2,\ \ldots,\ p;\ m=1,\ 2,\ \ldots,\ t).$$

Let the $p*p*t$ matrix

$$Y=B*_{lc}Z=B*_{lc}(A*_{lc}X)=B*_{lc}C \tag{24.23}$$

where

$$y_{ilm}=\sum_{j=1}^{p}(b_{ijl}z_{jlm})=\sum_{j=1}^{p}[b_{ijl}\sum_{k=1}^{p}(a_{jkl}x_{klm})]=\sum_{k=1}^{p}\sum_{j=1}^{p}[(b_{ijl}a_{jkl})x_{klm}] \quad (i=1, 2, \ldots, p) \qquad (24.24)$$

Extend (24.24) with respect to k, obtaining:

$$\begin{aligned}y_{ilm}=&[\sum_{j=1}^{p}(b_{ijl}a_{j1l})]x_{1lm}+[\sum_{j=1}^{p}(b_{ijl}a_{j2l})]x_{2lm}+\ldots+[\sum_{j=1}^{p}(b_{ijl}a_{ji-1l})]x_{i-1lm}\\&+[\sum_{j=1}^{p}(b_{ijl}a_{jil})]x_{ilm}+[\sum_{j=1}^{p}(b_{ijl}a_{ji+1l})]x_{i+1lm}+\ldots+[\sum_{j=1}^{p}(b_{ijl}a_{jpl})]x_{plm}\end{aligned} \qquad (24.25)$$

Let the coefficient of x_{klm} in (24.25)

$$\sum_{j=1}^{p}(b_{ijl}a_{jkl})=\begin{cases}1, & k=i\\ 0, & k\neq i\end{cases} \qquad (24.26)$$

then (24.25) becomes:

$$y_{ilm}=x_{ilm}\ (i, l=1, 2, \ldots, p;\ m=1, 2, \ldots, t), \qquad (24.27)$$

that is, (24.23) becomes:

$$Y=E_D*_{lc}X=X=B*_{lc}C. \qquad (24.28)$$

This is to say that (24.19) is solved. Extend (24.26), the condition for solution, with respect to *j*, obtaining:

$$\begin{cases}b_{i1l}a_{11l}+b_{i2l}a_{21l}+\ldots+b_{ipl}a_{p1l}=0\\ b_{i1l}a_{12l}+b_{i2l}a_{22l}+\ldots+b_{ipl}a_{p2l}=0\\ \cdots\cdots\cdots\cdots\\ b_{i1l}a_{1i-1l}+b_{i2l}a_{2i-1l}+\ldots+b_{ipl}a_{pi-1l}=0\\ b_{i1l}a_{1il}+b_{i2l}a_{2il}+\ldots+b_{ipl}a_{pil}=1\\ b_{i1l}a_{1i+1l}+b_{i2l}a_{2i+1l}+\ldots+b_{ipl}a_{pi+1l}=0\\ \cdots\cdots\cdots\cdots\\ b_{i1l}a_{1pl}+b_{i2l}a_{2pl}+\ldots+b_{ipl}a_{ppl}=0.\end{cases} \qquad (24.29)$$

The solution of (24.29) is:

$$b_{ijl}=\frac{\Delta b_{ijl}}{\Delta A_{Dl}^{T}} \quad (i, j, l=1, 2, \ldots, p) \qquad (24.20)$$

$$B=[b_{ijl}]_{ppp}=A^{-1}_{*lc*lc,\ ED,\ l},$$

where ΔA_{Dl}^{T} is the coefficient determinant of (24.29), Δb_{ijl} is the solution determinant of b_{ijl}.

The coefficient determinant

$$\Delta A_{Dl}^{T}=\begin{vmatrix}a_{11l} & a_{21l} & \cdots & a_{p1l}\\ a_{12l} & a_{22l} & \cdots & a_{p2l}\\ \cdots & \cdots & \cdots & \cdots\\ a_{1i-1l} & a_{2i-1l} & \cdots & a_{pi-1l}\\ a_{1il} & a_{2il} & \cdots & a_{pil}\\ a_{1i+1l} & a_{2i+1l} & \cdots & a_{pi+1l}\\ \cdots & \cdots & \cdots & \cdots\\ a_{1pl} & a_{2pl} & \cdots & a_{ppl}\end{vmatrix}$$

The planar matrix A^T_{Dl} of the determinant ΔA^T_{Dl} is the transpose of planar matrix of the lth depth plane A_{Dl} of the cubic matrix A. The solution determinant Δb_{ijl} is:

$$\Delta b_{ijl}=\Delta A^T_{Dl}\langle j\rangle \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix}^{\langle k=i\rangle} \tag{24.21}$$

The column vector

$$\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix}^{\langle k=i\rangle}$$

in (24.21) is such a vector that the element v_k in it is:

$$v_k=\begin{cases} 1, & k=i \\ 0, & k\neq i. \end{cases}$$

For example,

$$\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix}^{\langle k=2\rangle} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

$$\Delta A^T_{Dl}\langle j\rangle \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix}^{\langle k=i\rangle}$$

in (24.21) denotes the determinant obtained by substituting

$$\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_p \end{bmatrix}\langle k=i\rangle$$

for the *j*th column of ΔA^T_{Dl}. For example,

$$\Delta b_{213}=\Delta A^T_{D3}\langle 1\rangle \begin{vmatrix} 0 \\ 1 \\ 0 \\ \\ 0 \end{vmatrix} = \begin{vmatrix} 0 & a_{213} & \cdots & a_{p13} \\ 1 & a_{223} & \cdots & a_{p23} \\ 0 & a_{233} & \cdots & a_{p33} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & a_{2p3} & \cdots & a_{pp3} \end{vmatrix}.$$

ΔA^T_{Dl} in (24.20) is a divisor, there must be:

$\Delta A^T_{Dl}\neq 0$ (l=1, 2, …, p),

that is, A should be a full rank cubic matrix with respect to depth.

Now, we have found the left inverse $B=A^{-1}_{*lc*lc,\ ED,\ l}$ of A in the matrix equation (24.19). Since $*_{lc}$ in nonassociative with respect to $*_{lc}$, B is a nonassociative inverse.

Q.E.D.

Example 24.9: Suppose there is a cuboidal matrix equation:

$$A*_{lc}X=C \tag{24.19}$$

shown in Fig. 24.15. It is known that A is a full rank cubic matrix with respect to depth, find A^{-1}.

$$\begin{matrix} \begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{222} \end{bmatrix} \\ \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} \end{matrix} *lc \begin{bmatrix} x_{111} \\ x_{211} \end{bmatrix} \begin{bmatrix} x_{121} \\ x_{211} \end{bmatrix} = \begin{bmatrix} c_{111} \\ c_{211} \end{bmatrix} \begin{bmatrix} c_{121} \\ c_{221} \end{bmatrix}$$

Fig. 24.15 $A*_{lc}X=C$

Solution: B left $*lc$ multiplies the two sides of (24.19), obtaining:

$$B*_{lc}(A*_{lc}X)=B*_{lc}C. \tag{24.22}$$

The left side of (24.22) is shown in Fig. 24.16, the right side in Fig. 24.17.

In Fig. 24.16, $Z=A*_{lc}X$, $Y=B*_{lc}Z=B*_{lc}(A*_{lc}X)$. Let

$$
\begin{matrix}\begin{bmatrix} b_{111} & b_{121} \\ b_{211} & b_{221} \end{bmatrix} \\ \begin{bmatrix} b_{112} & b_{122} \\ b_{212} & b_{222} \end{bmatrix}\end{matrix}
\ast lc \left(\begin{matrix}\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} \\ \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}\end{matrix}
\ast lc \begin{bmatrix} x_{111} \\ x_{211} \end{bmatrix}\begin{bmatrix} x_{121} \\ x_{221} \end{bmatrix} \right)
$$

$$
= \begin{matrix}\begin{bmatrix} b_{111} & b_{121} \\ b_{211} & b_{221} \end{bmatrix} \\ \begin{bmatrix} b_{112} & b_{122} \\ b_{212} & b_{222} \end{bmatrix}\end{matrix}
\ast lc \begin{bmatrix} z_{111} = a_{111} x_{111} + a_{121} x_{211} \\ z_{211} = a_{211} x_{111} + a_{221} x_{211} \end{bmatrix}
\begin{bmatrix} z_{121} = a_{112} x_{121} + a_{122} x_{221} \\ z_{221} = a_{212} x_{121} + a_{222} x_{221} \end{bmatrix}
$$

$$
= \begin{bmatrix}
y_{111} = b_{111}(a_{111} x_{111} + a_{121} x_{211}) + b_{121}(a_{211} x_{111} + a_{221} x_{211}) = (b_{111} a_{111} + b_{121} a_{211}) x_{111} + (b_{111} a_{121} + b_{121} a_{221}) x_{211} \\
y_{211} = b_{211}(a_{111} x_{111} + a_{121} x_{211}) + b_{221}(a_{211} x_{111} + a_{221} x_{211}) = (b_{211} a_{111} + b_{221} a_{211}) x_{111} + (b_{211} a_{121} + b_{221} a_{221}) x_{211}
\end{bmatrix}
\begin{bmatrix}
y_{121} = b_{112}(a_{112} x_{121} + a_{122} x_{221}) + b_{122}(a_{212} x_{121} + a_{222} x_{221}) = (b_{112} a_{112} + b_{122} a_{212}) x_{121} + (b_{112} a_{122} + b_{122} a_{222}) x_{221} \\
y_{221} = b_{212}(a_{112} x_{121} + a_{122} x_{221}) + b_{222}(a_{212} x_{121} + a_{222} x_{221}) = (b_{212} a_{112} + b_{222} a_{212}) x_{121} + (b_{212} a_{122} + b_{222} a_{222}) x_{221}
\end{bmatrix}
$$

Fig. 24.16 Left side of $B *_{lc} (A *_{lc} X) = B *_{lc} C$

$$
\begin{matrix}\begin{bmatrix} b_{111} & b_{121} \\ b_{211} & b_{222} \end{bmatrix} \\ \begin{bmatrix} b_{112} & b_{122} \\ b_{212} & b_{222} \end{bmatrix}\end{matrix}
\ast lc \begin{bmatrix} c_{111} \\ c_{211} \end{bmatrix}\begin{bmatrix} c_{121} \\ c_{221} \end{bmatrix}
= \begin{bmatrix} x_{111} c_{111} + b_{121} c_{211} \\ b_{211} c_{111} + b_{221} c_{211} \end{bmatrix}
\begin{bmatrix} b_{112} c_{121} + b_{122} c_{221} \\ b_{212} c_{121} + b_{222} c_{221} \end{bmatrix}
$$

Fig. 24.17 Right side of $B *_{lc} (A *_{lc} X) = B *_{lc} C$

$$
\begin{cases}
b_{111}a_{111}+b_{121}a_{211}=1 \\
b_{111}a_{121}+b_{121}a_{221}=0 \\
b_{211}a_{111}+b_{221}a_{211}=0 \\
b_{211}a_{121}+b_{221}a_{211}=1 \\
b_{112}a_{112}+b_{122}a_{212}=1 \\
b_{112}a_{122}+b_{122}a_{222}=0 \\
b_{212}a_{112}+b_{222}a_{212}=0 \\
b_{212}a_{122}+b_{222}a_{222}=1
\end{cases} \tag{24.30}
$$

obtaining:

$$Y=E_D*_{lc}X=X^{To}=X=B*_{lc}C.$$

That is, equation (24.19) has been solved. Solve (24.30), obtaining:

$$B=[b_{ijl}]_{222}=A^{-1}{}_{*lc*lc,\ ED,\ l} \tag{24.31}$$

$$b_{ijl}=\frac{\Delta b_{ijl}}{\Delta A^{T}_{Dl}} \quad (i, j, l=1, 2) \tag{24.32}$$

where

$$\Delta A^{T}_{D1}=\begin{vmatrix} a_{111} & a_{211} \\ a_{121} & a_{221} \end{vmatrix} \qquad \Delta A^{T}_{D2}=\begin{vmatrix} a_{112} & a_{212} \\ a_{122} & a_{222} \end{vmatrix}$$

$$\Delta b_{111}=\Delta A_{D1}{}^{T}\langle 1\rangle\begin{vmatrix} V_1 \\ V_2 \end{vmatrix}\langle k=1\rangle=\begin{vmatrix} 1 & a_{211} \\ 0 & a_{221} \end{vmatrix} \qquad \Delta b_{121}=\begin{vmatrix} a_{111} & 1 \\ a_{121} & 0 \end{vmatrix}$$

$$\Delta b_{211}=\begin{vmatrix} 0 & a_{211} \\ 1 & a_{221} \end{vmatrix} \qquad \Delta b_{221}=\begin{vmatrix} a_{111} & 0 \\ a_{121} & 1 \end{vmatrix}$$

$$\Delta b_{112}=\begin{vmatrix} 1 & a_{212} \\ 0 & a_{222} \end{vmatrix} \qquad \Delta b_{122}=\begin{vmatrix} a_{112} & 1 \\ a_{122} & 0 \end{vmatrix}$$

$$\Delta b_{212}=\begin{vmatrix} 0 & a_{212} \\ 1 & a_{222} \end{vmatrix} \qquad \Delta b_{222}=\begin{vmatrix} a_{112} & 0 \\ a_{122} & 1 \end{vmatrix}$$

$$b_{111}=\frac{\Delta b_{111}}{\Delta A^{T}_{D1}} \ldots\ldots b_{222}=\frac{\Delta b_{222}}{\Delta A^{T}_{D2}}.$$

Q.E.D.

The proofs of the following theorems are similar to that of Theorem 24.3, so are all omitted. These theorems say that (24.17) and (24.18) are solvable for both $*_{lc}$ and $*_{rc}$.

Theorem 24.4: Suppose $A=[a_{jkl}]_{ppp}$, $X=[x_{klm}]_{ppt}$, $C=[c_{jlm}]_{ppt}$, then if and only if A is full rank with respect to width, then A in

$$A*_{rc}X=C$$

has an associative inverse:

$$B=[b_{ijk}]_{ppp}=A^{-1}{}_{*lc*rc,\ EW,\ l}$$

where

$$b_{ijk}=\frac{\Delta b_{ijk}}{\Delta A^{T}_{Wk}}$$

$$\Delta b_{ijk}=\Delta A^{T}_{Wk}\langle j\rangle\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_l \\ \cdot \\ \cdot \\ v_p \end{bmatrix}\langle k=i\rangle$$

Theorem 24.5: Suppose $Y=[y_{ijk}]_{pqq}$, $A=[a_{jkl}]_{qqq}$, $C=[c_{ijl}]_{pqq}$, then if and only if A is full rank with respect to height, then A in

$Y*_{rc}A=C$

has a nonassociative inverse:

$B=[b_{jlm}]_{qqq}=A^{-1}_{*rc*rc,\ EH,\ r}$

where

$$b_{jlm}=\frac{\Delta b_{jlm}}{\Delta A^{T}_{Hj}}$$

$$\Delta b_{jlm}=\Delta A^{T}_{Hj}\langle 1\rangle\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_k \\ \cdot \\ \cdot \\ v_q \end{bmatrix}\langle k=m\rangle$$

Theorem 24.6: Suppose $Y=[y_{ijk}]_{pqq}$, $A=[a_{jkl}]_{qqq}$, $C=[c_{ijl}]_{pqq}$, then if and only if A is full rank with respect to width, then A in

$Y*_{lc}A=C$

has an associative inverse:

$B=[b_{klm}]_{qqq}=A^{-1}_{*rc*lc,\ EW,\ r}$

where

$$b_{klm}=\frac{\Delta b_{klm}}{\Delta A^{T}_{Wk}}$$

$$\Delta b_{klm}=\Delta A^{T}_{WK}\langle 1\rangle\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_j \\ \cdot \\ \cdot \\ v_q \end{bmatrix}\langle j=m\rangle$$

24.4 Tensor matrices

24.4.1 Matrix representation of tensors

Definition 24.24: An nth order m-dimensional tensor is a quantity consisting of m^n components $a_{k1k1\ldots kn}$ ($k_1, k_2, \ldots, k_n$=1, 2, ..., m), which transform under a change of coordinate system $x_i'=\alpha_{ij}x_j$ according to the law

$$a'_{i1i2\ldots in}=\alpha_{i1k1}\alpha_{i2k2}\ldots\alpha_{inkn}a_{k1k2\ldots kn} \tag{24.33}$$

where $a_{k1k2\ldots kn}$ and $a_{i1i2\ldots in}'$ are the components of the tensor with respect to the coordinate systems x_j and x_i' respectively and α_{ijkj} is the cosine of the angle between the x_{ij}'-axis and the x_{kj}-axis for i_j, k_j=1, 2, ..., m.

Let the quantity mentioned above be denoted by an n-dimensional order m matrix $[a_{k1k2\ldots kn}]$, let the coordinate transformations $\alpha_{i1k1}, \alpha_{i2k2}, \ldots, \alpha_{inkn}$ be denoted by 2-dimensional order m matrices $[\alpha_{i1k1}], [\alpha_{i2k2}], \ldots, [\alpha_{inkn}]$, then (24.33) can be denoted by the inner products of matrices:

$$[a_{i1i2\ldots in}']=\{[\alpha_{i1k1}]*_i^{k1}\{[\alpha_{i2k2}]*_i^{k2}\{\ldots\{[\alpha_{inkn}]*_i^{kn}[a_{k1k2\ldots kn}]\}\ldots\}\} \tag{24.34}$$

$[a_{k1k2\ldots kn}]$ satisfying (24.34) is called a tensor matrix.

Example 24.10: Let n=3, m=2, then (24.33) becomes (24.35)

$$a_{ijk}'=\alpha_{ip}\alpha_{jq}\alpha_{kr}a_{pqr} \tag{24.35}$$

and (24.34) becomes (24.36) shown in Fig. 24.18.

$$[a_{ijk}']=\{[\alpha_{ip}]*_i^p\{[\alpha_{jq}]*_i^q\{[\alpha_{kr}]*_i^r[a_{pqr}]\}\}\} \tag{24.36}$$

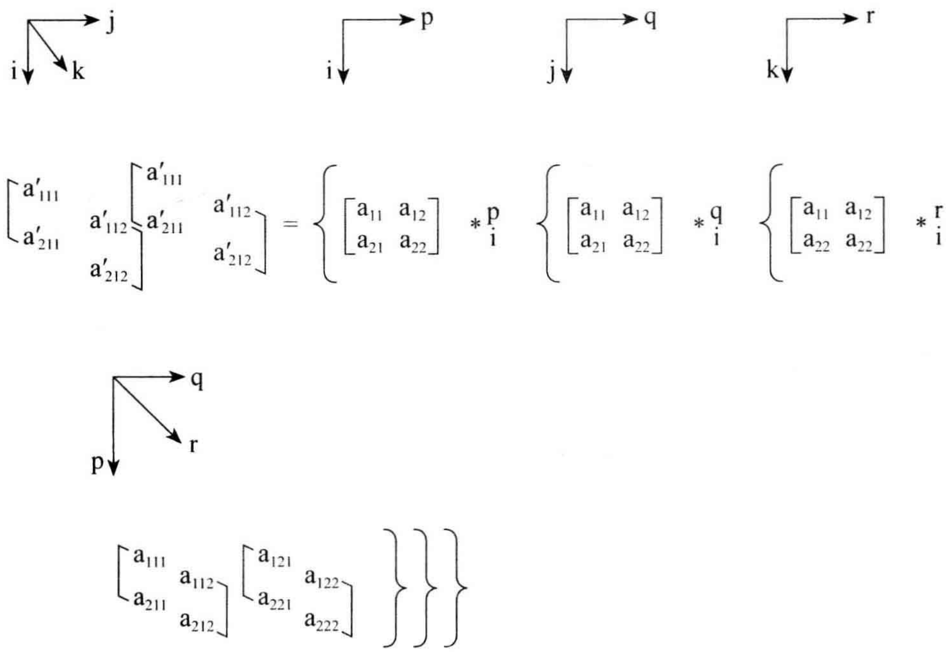


Fig. 24.18 $[a_{ijk}']=\{[\alpha_{ip}]*_i^p\{[\alpha_{jq}]*_i^q\{[\alpha_{kr}]*_i^r[a_{pqr}]\}\}\}$

24.4.2 Tensor matrix algebra

Since a tensor is denoted by a tensor matrix, tensor algebra becomes tensor matrix algebra.

24.4.2.1 Addition of tensor matrices

Two tensor matrices of the same dimension and order can be added, similar to the addition of cuboidal matrices.

24.4.2.2 Outer product of tensor matrices

The definition of the outer product of tensor matrices is similar to Definition 24.4.

Example 24.11: Suppose $A=[a_i]_2=[a_1, a_2]$ is a 1-dimensional order 2 tensor matrix, $B=[b_{jk}]_{22}=\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ is a 2-dimensional order 2 tensor matrix, then the outer product of A and B is $C=A*_oB$ shown in Fig. 24.19.

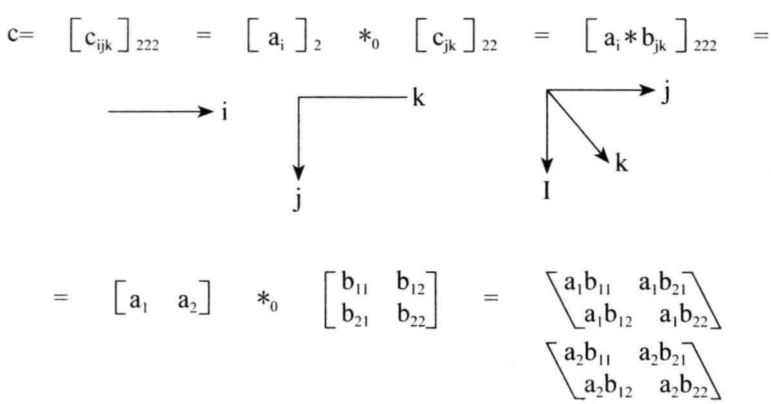


Fig. 24.19 Outer product of tensor matrices

24.4.2.3 Contractions and inner products of tensor matrices

Definition 24.25: The operation of setting two of the indices in the components of a r-dimensional tensor matrix ($r \geqslant 2$) equal and then summing with respect to the repeated index is called contraction.

Example 24.12: Consider the components a_{ij} of a 2-dimensional tensor matrix. The contraction of this tensor matrix yields the scalar

$a_{ii}=a_{11}+a_{22}+a_{33}$.

Definition 24.26: An inner product of two tensor matrices A and B is a contraction of the outer product with respect to two indices, each belonging to a component of the tensor matrices, denoted by $C=A*_iB$.

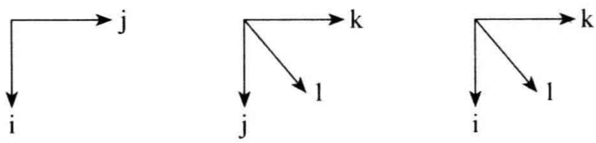
Definition 24.26 tells us that in making inner product of two tensor matrices A and B, we first make outer product of A and B, and then make contraction. But in practice, we first set one index in the components of A and one index in the components of B equal, and then make inner product of A and B according to Definition 24.1.

Example 24.13: Suppose

$$A=[a_{ij}]_{22}=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B=[b_{jkl}]_{222}=\begin{matrix} \begin{bmatrix} b_{111} & b_{121} \\ b_{112} & b_{122} \end{bmatrix} \\ \begin{bmatrix} b_{211} & b_{221} \\ b_{212} & b_{222} \end{bmatrix} \end{matrix}$$

then the inner product of A and B, $C=[c_{ikl}]_{222}=A*_iB=[a_{ij}]_{22}*_i[b_{jkl}]_{222}=[\,]_{222}$, is shown in Fig. 24.20.



$$= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} *i \begin{matrix} \begin{bmatrix} b_{111} & b_{121} \\ b_{112} & b_{122} \end{bmatrix} \\ \begin{bmatrix} b_{211} & b_{221} \\ b_{212} & b_{222} \end{bmatrix} \end{matrix} = \begin{matrix} \begin{bmatrix} c_{111}= & c_{121}= \\ a_{11}b_{111}a_{12}b_{211} & a_{11}b_{121}a_{12}b_{221} \\ c_{111}= & c_{122}= \\ a_{11}b_{112}a_{12}b_{212} & a_{11}b_{122}a_{12}b_{222} \end{bmatrix} \\ \begin{bmatrix} c_{211}= & c_{221}= \\ a_{21}b_{111}a_{22}b_{211} & a_{21}b_{121}a_{22}b_{221} \\ c_{212}= & c_{222}= \\ a_{21}b_{112}a_{22}b_{212} & a_{21}b_{122}a_{22}b_{222} \end{bmatrix} \end{matrix}$$

Fig. 24.20 Inner product of tensor matrices

Part 11
Applications of decomposition

Hypothetical inference is divided into recessive hypothetical inference and dominant hypothetical inference. Rule-based systems are divided into inference rule systems and mutually-inversistic decomposition systems. In inference rule systems, all of the rules are treated as inference rules. In mutually-inversistic decomposition systems, there are only 5 inference rules, the other rules are treated as axioms or theorems. Mutually-inversistic decomposition systems have been studied in Part 1, this part studies mutually-inversistic automated decomposition systems, which are composed of first-level automated decomposition systems and second-level automated decomposition systems. Mutually-inversistic second-level automated decomposition systems can be obtained in three ways: lifting, transforming, and bringing-into. Lifting refers to that a mutually-inversistic first-level automated decomposition is lifted to a mutually-inversistic second-level automated decomposition system; e.g., Prolog is lifted to second-level single quasi-Prolog. Transforming refers to that a second-level inference rule system is transformed into a mutually-inversistic second-level automated decomposition system; e.g., Hoare rules of program verification are transformed into mutually-inversistic program verification. Bringing-into refers to that an axiomatic system with only one inference rule: rule of detachment, is brought into a mutually-inversistic second-level automated decomposition system. This part applies decomposition to relational database, planning and scheduling, semantic network, expert systems, program verification, natural language processing, hardware verification, automated theorem proving, etc..

Chapter 25
Inference rule systems vs. mutually-inversistic automated decomposition systems

25.1 Recessive hypothetical inference vs. dominant hypothetical inference

25.1.1 First-level recessive hypothetical inference vs. first-level dominant hypothetical inference

The first-level affirmative expression of hypothetical inference is as follows:

sentence→noun, verb

sentence

----

noun, verb.

Here, the first row above the line is the major premise: sentence→noun, verb; sentence is its antecedent, noun, verb is its consequent. The second row above the line is the minor premise: sentence. The row below the line is the conclusion: noun, verb. The inference is made as follows: the minor premise "sentence" unifies successfully with the antecedent "sentence" of the major premise "sentence→noun, verb", and the conclusion "noun, verb" is detached.

The first-level negative expression of hypothetical inference is as follows:

sentence→noun, verb

¬(noun, verb)

¬(sentence).

Here, the second row above the line is the negation of the conclusion: ¬(noun, verb). The row below the line is the negation of the minor premise: ¬(sentence). The inference is made as follows: the negation of the conclusion: ¬(noun, verb) unifies successfully with the consequent of the major premise: noun, verb, and the negation of the minor premise: ¬(sentence) is inferred.

When a human being makes the first-level affirmative expression of hypothetical inference, the unification he or she makes is intangible, and can be omitted, thus the

expression is simplified as the inference rule:

sentence

----

noun, verb.

This is the first-level recessive affirmative expression of hypothetical inference. See Example 25.1.

Example 25.1: Suppose we have the following grammar rules:

sentence	noun	verb
-------------	------	-------
noun, verb	John	reads.

Derive "John reads" from "sentence" manually.

Solution: sentence$\Rightarrow$noun, verb$\Rightarrow$John, verb$\Rightarrow$John, reads.

Take the first step of the derivation sentence$\Rightarrow$noun, verb as an example. Because there is an inference rule:

sentence

----

noun, verb,

so, we derive "noun, verb" from "sentence" directly.

Recessive hypothetical inference is concise, many mathematical proofs are done with it.

When a computer makes the first-level negative expression of hypothetical inference, the unification it makes is tangible, and cannot be omitted. This is the first-level dominant negative expression of hypothetical inference. See Example 25.2.

Example 25.2: Suppose we have the following Prolog program and goal.

(1) sentence(s_0, s):-noun(s_0, s_1),verb(s_1, s).

(2) noun([John|s], s).

(3) verb([reads|s], s).

(4) ?-sentence([John, reads], []).

The SLD tree of the program and goal is shown in Fig. 25.1.

In Fig. 25.1, ?-sentence([John, reads], []), the initial goal, is the negation of the conclusion. Sentence(s_0, s), the head of the conditional clause (1), is the consequent of the major premise. ?-sentence([John, reads], []) unifies with sentence(s_0, s) successfully, and ?-noun([John, reads], s_1), verb(s_1, []) is inferred. The unification is tangible, and cannot be omitted. This is the first-level dominant negative expression of hypothetical inference.

Dominant hypothetical inference is just hypothetical inference.

?-sentence([John, reads], []).

(1) $\{s_0/[\text{John, reads}], s/[]\}$

?-noun([John, reads], s_1), verb(s_1, []).

(2) $\{s/\text{reads}, s/s_1\}$

?-verb([reads], []).

(3) $\{s/[]\}$

□

Fig. 25.1 SLD tree for Example 25.2

25.1.2 Second-level recessive hypothetical inference vs. second-level dominant hypothetical inference

Take "all integers are rationals, all rationals are real numbers, therefore, all integers are real numbers" as an example. The second-level affirmative expression of hypothetical inference is as follows:

$$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$$
$$\{\text{int}(x)\leqslant^{-1}\text{rat}(x)\}\wedge\{\text{rat}(x)\leqslant^{-1}\text{real}(x)\}$$

$$\text{int}(x)\leqslant^{-1}\text{real}(x).$$

Here, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ is the major premise, $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ and $P\leqslant^{-1}R$ are the antecedent and consequent of the major premise respectively. $\{\text{int}(x)\leqslant^{-1}\text{rat}(x)\}\wedge\{\text{rat}(x)\leqslant^{-1}\text{real}(x)\}$ is the minor premise. $\text{Int}(x)\leqslant^{-1}\text{real}(x)$ is the conclusion. The inference goes as follows: when P is assigned int(x), Q rat(x), R real(x), the minor premise $\{\text{int}(x)\leqslant^{-1}\text{rat}(x)\}\wedge\{\text{rat}(x)\leqslant^{-1}\text{real}(x)\}$ unifies with the antecedent of the major premise $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}$ successfully, and the conclusion $\text{int}(x)\leqslant^{-1}\text{real}(x)$ is detached.

The second-level negative expression of hypothetical inference is as follows:

$$\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$$
$$\neg\{\text{int}(x)\leqslant^{-1}\text{real}(x)\}$$

$$\neg\{\{\text{int}(x)\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}\text{real}(x)\}\}.$$

Here, $\neg\{\text{int}(x)\leqslant^{-1}\text{real}(x)\}$ is the negation of the conclusion, $\neg\{\{\text{int}(x)\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}\text{real}(x)\}\}$ is the negation of the minor premise. The inference goes as follows: when P is

assigned int(x), R real(x), the negation of the conclusion $\neg\{\text{int}(x)\leqslant^{-1}\text{real}(x)\}$ unifies with the consequent of the major premise $\{P\leqslant^{-1}R\}$ successfully, and the negation of the minor premise $\neg\{\{\text{int}(x)\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}\text{real}(x)\}\}$ is inferred.

When a human being makes the second-level affirmative expression of hypothetical inference, the unification he or she makes is intangible, and can be omitted. Thus, the expression is simplified as the inference rule:

$P\leqslant^{-1}Q$

$Q\leqslant^{-1}R$

$P\leqslant^{-1}R$,

called hypothetical syllogism. According to the inference rule, the human being infers int(x)$\leqslant^{-1}$real(x) from int(x)$\leqslant^{-1}$rat(x) and rat(x)$\leqslant^{-1}$real(x) directly. This is the second-level recessive affirmative expression of hypothetical inference.

When a computer makes the second-level negative expression of hypothetical inference, the unification it makes is tangible, and cannot be omitted. This is the second-level dominant negative expression of hypothetical inference, see Example 14.7. In this example, the unification of ?-{real(x)==>>int(x)}, the negation of the conclusion, and {R==>>P}, the consequent of the major premise, is tangible, and cannot be omitted.

25.2 Inference rule systems vs. mutually-inversistic automated decomposition systems

25.2.1 Inference rule systems

Rule-based systems are divided into inference rule systems and mutually-inversistic decomposition systems. Inference rule systems are based on recessive hypothetical inference. They treat rules as inference rules. There are many inference rules in an inference rule system. It is skillful when to choose which inference rule. It is easy for a human being but hard for a computer to choose from these inference rules. Therefore, inference rule systems are suitable for human beings. Manual parser of Example 25.1 is an example of first-level inference rule system. Hoare rules of program verification, where there are 5 inference rules, is an example of second-level inference rule system.

25.2.2 Mutually-inversistic automated decomposition systems

Mutually-inversistic decomposition systems are based on dominant hypothetical inference. They have only 5 inference rules: the affirmative expression of hypothetical inference (also called the rule of detachment and modus ponens), the negative expression

of hypothetical inference (also called modus tollens), the disjunctive inference, the rule of conjunction, the rule of change from $\vee$ to $\vee/^{-1}$. The other rules are treated as axioms or theorems.

Mutually-inversistic decomposition systems are divided into forward systems and backward systems. A forward system has only two inference rules: the affirmative expression of hypothetical inference and the rule of conjunction. If it has only one inference rule: the affirmative expression of hypothetical inference, then it is an automated system. Because there is only one inference rule, every time the system makes an inference, the system employs it. As to the numerous axioms or theorems, the system searches them top-down. This process is mechanical, suitable for a computer to do. Mutually-inversistic hardware verifier is an example of mutually-inversistic first-level forward automated decomposition system. A mathematical axiomatic system with only one inference rule, the rule of detachment, is an example of mutually-inversistic second-level forward automated decomposition system.

A backward automated system has only 3 inference rules: the negative expression of hypothetical inference, the disjunctive inference, the rule of change from $\vee$ to $\vee/^{-1}$. Prolog is an example of mutually-inversistic first-level backward automated decomposition system. Second-level single quasi-Prolog is an example of mutually-inversistic second-level backward automated decomposition system. Why are the systems still automated despite the three inference rules? Because in any situation, only one inference rule can be chosen. Take Prolog as an example. If the leftmost subgoal unifies with a conditional clause, then the rule of negative expression of hypothetical inference is chosen. For example, in Fig. 25.1, the leftmost subgoal ?-sentence([John, reads], []) unifies with the conditional clause (1), and the rule of negative expression of hypothetical inference is chosen. If the body of the conditional clause inferred is the disjunction of two or more atoms, then the rule of change from $\vee$ to $\vee/^{-1}$ is chosen. For example, in Fig. 25.1, the body of the conditional clause inferred ?-noun([John, reads], s_l),verb(s_l, []), that is $\neg$noun([John, reads], s_l)$\vee\neg$verb(s_l, []) is the disjunction of two atoms, and it is changed to $\neg$noun([John, reads], s_l)$\vee/^{-1}$ $\neg$verb(s_l, []). If the leftmost subgoal unifies with an unconditional clause, then the rule of disjunctive inference is chosen. For example, in Fig. 25.1, the leftmost subgoal ?-noun([John, reads], s_l),verb(s_l, []) unifies with the unconditional clause (2), and the rule of disjunctive inference is chosen.

A forward inference rule system can be transformed into a mutually-inversistic forward automated decomposition system by transforming the rules from inference rules to axioms or theorems, taking the affirmative expression of hypothetical inference as the sole inference rule. For example, Hoare rules of program verification are transformed into mutually-inversistic program verification in this way.

25.2.3 Theoretically inference rule systems, implementally mutually-inversistic automated decomposition systems

An inference rule system is suitable for a human being, a mutually-inversistic automated decomposition system is suitable for a computer. But some people would say: "many inference rule systems have been implemented on computers, some even automated." This is true. The reason is that when implementing on a computer, the inference rules in these inference rule systems are implemented by the rule clauses of Prolog or the if statement of C++, which implement them as theorems. So, from the theoretical perspective, these systems are inference rules systems, but from the implemental perspective, they are mutually-inversistic automated decomposition systems. For example, in Example 14.7, the inference rule hypothetical syllogism is implemented by a second-level rule clause, and it is a logical theorem there. An example of an if statement denoting the major premise of a dominant hypothetical inference, see Example 25.3.

Example 25.3: Convert upper case English letters into lower case ones.

The C program for conversion is as follows:

```
main()
{char ch;
scanf ("%c", &ch);
if (ch>='A'&&ch<='Z') printf ("%c", ch+32);
}
```

where 32 is the difference of the positions of upper case letters and lower case letters in the ASCII table. When a B is inputted, a b is outputted.

After the B is inputted, the ch in the scanf statement is assigned B, the ch of the condition ch>='A'&&ch<='Z' in the if statement is also assigned the B, and 'B'>='A'&&'B'<='Z' is computed, the result is "true", then the printf statement is activated, the b is printed. It is tangible that the ch of the condition ch>='A'&&ch<='Z' in the if statement being assigned B results in'B'>='A'&&'B'<='Z' being computed. For example, we can assemble the program into the object codes of the assembly language, and enter the single step trace mode, trace the execution of the program step by step. Therefore, the process is a dominant hypothetical inference, denoted by:

```
if (ch>='A'&&ch<='Z') printf ("%c", ch+32)
'B'>='A'&&'B'<='Z'
-------------------------------------------------------
                    b.
```

And the if statement denotes the major premise of the dominant hypothetical inference.

Chapter 26 Mutually-inversistic relational databases

26.1 First-level single quasi-relational databases

Example 26.1: The known: turn_around(*planet*, *fixed_star*) denoted by Table 26.1, turn_around(*satellite*, *planet*) denoted by Table 26.2, and the single empirical or mathematical theorem turn_around(*planet*, *fixed_star*) ∩ turn_around(*satellite*, *planet*) $\subseteq^{-1}$ turn_around(*satellite*, *fixed_star*). And we want to query: turn_around(Moon, Sun).

Table 26.1 Turn_around (*planet, fixed_star*)

planet	*fixed_star*
Earth	Sun
Jupiter	Sun

Table 26.2 Turn_around (*satellite, planet*)

satellite	*planet*
Moon	Earth
Jupiter_satel_one	Jupiter

Solution: We use indirect method of proof. We want to query turn_around(Moon, Sun), then we suppose that ~turn_around(Moon, Sun) holds. We make the negative expression of hypothetical inference from it and the empirical or mathematical theorem turn_around (*planet*, *fixed_star*) ∩ turn_around(*satellite*, *planet*) $\subseteq^{-1}$ turn_around(*satellite*, *fixed_star*), inferring

~{turn_around(*planet*, Sun) ∩ turn_around(Moon, *planet*)} (26.1)

Formula (26.1) reminds us to make the fact second intersection of turn_around(*planet*, Sun) (Table 26.1)and turn_around(Moon, *planet*) (Table 26.2), shown in Table 26.3.

Table 26.3 turn_around(*satellite, fixed_star*)

satellite	*fixed_star*
Moon	Sun
Jupiter_satel_one	Sun

Formula (26.1) says that the fact second intersection of turn_around(*planet*, Sun) and turn_around(Moon, *planet*) does not exist, while Table 26.3 says that it exists, thus, contradiction occurs. This shows that the supposition ~turn_around(Moon, Sun) is false, and we prove (query) turn_around(Moon, Sun). The SQL statements of the query is as follows:

SELECT *satellite*, *fixed_star*

FROM turn_around(*planet*, *fixed_star*), turn_around(*satellite*, *planet*)

WHERE *satellite*=Moon, *fixed_star*=Sun.

26.2 Second-level single quasi-relational databases

The design of the second-level single quasi-relational database should be that no table is redundant, and that query is flexible and efficient, capable of querying more information. In order for each table being not redundant, we do the following.

$$\{P\cup|^{-1}Q\}=^{-1}\{\sim P\subseteq^{-1}Q\}$$
$$\{P\cup^{-1}Q\}=^{-1}\{\sim P\subset^{-1}Q\}$$
$$\{P\oplus^{-1}Q\}=^{-1}\{\sim P=^{-1}Q\}.$$

Because of the above formulas, we can transform $\cup|^{-1}$, $\cup^{-1}$, and $\oplus^{-1}$ into $\subseteq^{-1}$, $\subset^{-1}$, and $=^{-1}$ respectively. Therefore , in the construction of the second-level single quasi-relational database, we need only to consider $|\cap^{-1}$, $\subseteq^{-1}$, $=^{-1}$, $\subset^{-1}$, and $\times^{-1}$.

Some empirical or mathematical connection operators are strong, others are weak. For example,

$$\{P\subset^{-1}Q\}\subseteq^{-1}\{P\subseteq^{-1}Q\}$$

tells us that from $P\subset^{-1}Q$, $P\subseteq^{-1}Q$ can be inferred. Therefore, $P\subset^{-1}Q$ is stronger than $P\subseteq^{-1}Q$. If two fact propositions satisfy a stronger empirical or mathematical connection operator, then there is no need to preserve a weaker one, because the weaker one can be inferred from the stronger one. For example, if int(x)$\subset^{-1}$rat(x) holds, then there is no need to preserve int(x)$\subseteq^{-1}$rat(x), because the latter can be inferred from the former. Other single set theorems reflecting the strongness of empirical or mathematical connection operators are

$$\{P=^{-1}Q\}\subseteq^{-1}\{P\subseteq^{-1}Q\}$$
$$\{P\subseteq^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$$
$$\{P\times^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}.$$

According to these theorems, we can make a figure reflecting the strongness of the empirical or mathematical connection operators shown in Fig. 26.1.

In Fig. 26.1, the three unicellular operators at the top are the strongest, $\subseteq^{-1}$ in the middle is intermediate, $|\cap^{-1}$ at the bottom is the weakest. We only consider the construction of $\subset^{-1}$ table, $=^{-1}$ table, and $\times^{-1}$ table.

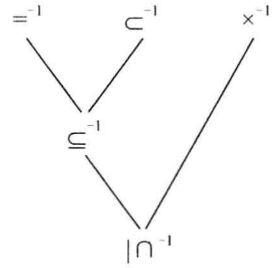


Fig. 26.1 Strongness of empirical or mathematical connection operators

There exist $\subset^{-1}$ connections among int(x), rat(x), and real(x), shown in Fig. 26.2, where the arrows show the directions of function determinations; i.e., int(x) functionally determines rat(x), and rat(x) functionally determines real(x).

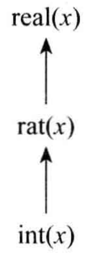


Fig. 26.2 $\subset^{-1}$ connections

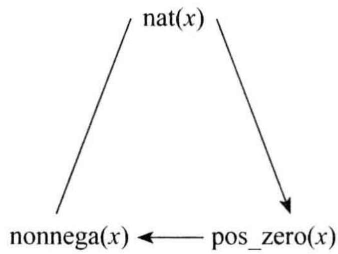


Fig. 26.3 $=^{-1}$ connections

According to Fig. 26.2, we can make the $P\subset^{-1}Q$ table shown in Table 26.4, where P is the primary key.

Table 26.4 $P\subset^{-1}Q$

P	Q
int(x)	rat(x)
rat(x)	real(x)

Table 26.5 $P=^{-1}Q$

P	Q
nat(x)	pos_zero(x)
pos_zero(x)	nonnega(x)

Table 26.4 only contains the two $\subset^{-1}$ connections indicated by the arrows of Fig. 26.2, it does not contain int(x)$\subset^{-1}$real(x). This is because $\subset^{-1}$ satisfies transitive law, int(x)$\subset^{-1}$ real(x) can be inferred from the two $\subset^{-1}$ connections indicated by the arrows of Fig. 26.2. Thus, Table 26.4 eliminates the transitive dependency, satisfying BC normal form.

There exist $=^{-1}$ connections among nat(x), pos_zero(x) (positive integers and zero), and nonnega(x) (nonnegative integers), shown in Fig. 26.3, where the arrows show the directions of function determinations. According to Fig. 26.3, we can make the $P=^{-1}Q$ table shown in Table 26.5, where P is the primary key. Table 26.5 only contains the two $=^{-1}$

connections indicated by the arrows of Fig. 26.3, it does not contain nonnega(x)$=^{-1}$nat(x). This is because $=^{-1}$ satisfies transitive law, nonnega(x)$=^{-1}$nat(x) can be inferred from the two $=^{-1}$ connections indicated by the arrows of Fig. 26.3. Thus, Table 26.5 eliminates the transitive dependency, satisfying BC normal form.

If a partition of a concept contains only two parts, then the partition is called a binary partition. Two different binary partitions of a concept satisfy $\times^{-1}$ connection. For example, a number is divided into a positive number (pos) and a nonpositive number (nonpos), it is also divided into a negative number (nega) and a nonnegative number (nonnega). Thus, pos(x)$\times^{-1}$nega(x) holds. $\times^{-1}$ connections are shown in Fig. 26.4, where the arrows show the directions of function determinations.

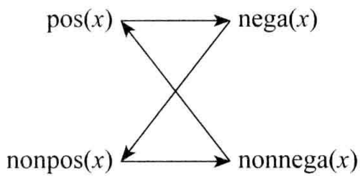


Fig. 26.4 $\times^{-1}$ **connections**

According to Fig. 26.4, we make $P\times^{-1}Q$ table shown in Table 26.6, where P is the primary key. Table 26.6 satisfies BC normal form.

Table 26.6 $P\times^{-1}Q$

P	Q
pos(x)	nega(x)
nega(x)	nonpos(x)
nonpos(x)	nonnega(x)
nonnega(x)	pos(x)

Example 26.2: Suppose we have $P\times^{-1}Q$ table shown in Table 26.7 and $Q\subset^{-1}R$ table shown in Table 26.8. We want to query pos(x)$|\cap^{-1}$rat(x).

Table 26.7 $P\times^{-1}Q$

P	Q
pos(x)	int(x)
int(x)	

Table 26.8 $Q\subset^{-1}R$

Q	R
int(x)	rat(x)
rat(x)	real(x)
real(x)	

Solution: First, let us carry out integrity analysis to Tables 26.7 and 26.8. Table 26.7 satisfies entity integrity, its primary key P does not adopt null value. Likewise, Table 26.8 satisfies entity integrity. Table 26.7 satisfies referential integrity, its foreign key Q adopts either the value of the primary key Q of the $Q\subset^{-1}R$ table (say, int(x)), or null value. Likewise, Tables 26.7 and 26.8 satisfy referential integrity themselves.

We use the indirect proof method to query. In order to query pos(x)$|\cap^{-1}$rat(x), we suppose $\sim$$\{$pos($x$)$|\cap^{-1}$rat($x$)$\}$ to be true. $\sim$$\{$pos($x$)$|\cap^{-1}$rat($x$)$\}$ and $\{P|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P|\cap^{-1}R\}$ make the second-level negative expression of hypothetical inference, inferring

$$\sim\{\{\mathrm{pos}(x)|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}\mathrm{rat}(x)\}\} \tag{26.2}$$

Formula (26.2) is equivalent to

$$\sim\{\mathrm{pos}(x)|\cap^{-1}Q\}\cup\sim\{Q\subseteq^{-1}\mathrm{rat}(x)\} \tag{26.3}$$

Formula (26.3) can be changed to

$$\sim\{\mathrm{pos}(x)|\cap^{-1}Q\}\cup|^{-1}\sim\{Q\subseteq^{-1}\mathrm{rat}(x)\} \tag{26.4}$$

Formula (26.4) reminds us to query $Q\subseteq^{-1}$rat(x) table and pos(x)$|\cap^{-1}Q$ table respectively.

First, we query $Q\subseteq^{-1}$rat(x) table. There is no $\subseteq^{-1}$ table. But, $\sim$$\{Q\subseteq^{-1}$rat($x$)$\}$ and $\{P\subset^{-1}Q\}\subseteq^{-1}\{P\subseteq^{-1}Q\}$ make the second-level negative expression of hypothetical inference, inferring $\sim$$\{Q\subset^{-1}$rat($x$)$\}$. $\sim$$\{Q\subset^{-1}$rat($x$)$\}$ reminds us to query the $Q\subset^{-1}$rat(x) table. As expected, in Table 26.8, we query int(x)$\subset^{-1}$rat(x). Int(x)$\subset^{-1}$rat(x) and $\{Q\subset^{-1}R\}\subseteq^{-1}\{Q\subseteq^{-1}R\}$ make the second-level affirmative expression of hypothetical inference, inferring int(x)$\subseteq^{-1}$rat(x). Thus, we can make $Q\subseteq^{-1}R$ table shown in Table 26.9.

Table 26.9 $\boldsymbol{Q\subseteq^{-1}R}$

Q	R
int(x)	rat(x)

Table 26.10 $\boldsymbol{P|\cap^{-1}Q}$

P	Q
pos(x)	int(x)

Then, we query pos(x)$|\cap^{-1}Q$ table. There is no $|\cap^{-1}$ table. But, $\sim$$\{$pos($x$)$|\cap^{-1}Q\}$ and $\{P\times^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$ make the second-level negative expression of hypothetical inference, inferring $\sim$$\{$pos($x$)$\times^{-1}Q\}$. $\sim$$\{$pos($x$)$\times^{-1}Q\}$ reminds us to query pos(x)$\times^{-1}Q$ table. As expected, in Table 26.7, we query pos(x)$\times^{-1}$int(x). pos(x)$\times^{-1}$int(x) and $\{P\times^{-1}Q\}\subseteq^{-1}\{P|\cap^{-1}Q\}$ make the second-level affirmative expression of hypothetical inference, inferring pos(x)$|\cap^{-1}$ int(x). Thus, we can make $P|\cap^{-1}Q$ table shown in Table 26.10.

Formula (26.2) says that the empirical or mathematical second intersection of pos(x)$|\cap^{-1}Q$ and $Q\subseteq^{-1}$rat(x) does not exist, which reminds us to make the empirical or mathematical second intersection of Tables 26.10 and 26.9, obtaining Table 26.11.

Table 26.11 $\{\text{pos}(x)|\cap^{-1}Q\}\cap\{Q\subseteq^{-1}\text{rat}(x)\}$

P	R
pos(x)	rat(x)

Formula (26.2) says that the empirical or mathematical second intersection of pos(x)$|\cap^{-1}Q$ and $Q\subseteq^{-1}$rat(x) does not exist, while Table 26.11 shows that it exists. Thus, contradiction occurs. This shows the supposition $\sim$\{pos(x)$|\cap^{-1}$rat(x)\} is false, and we prove (query) pos(x)$|\cap^{-1}$rat(x). The SQL statements of the query is as follows:

SELECT P, R

FROM $P|\cap^{-1}Q$, $Q\subseteq^{-1}R$

WHERE P=pos(x), R=rat(x).

In the query, we can use the second-order main. For example, we can query int(x) $\subset^{-1}r(x)$; i.e., we ask what can be inferred from int(x). The answer is r=rat and r=real. In the query, we can also use the third-order main. For example, we can query int(x)φreal(x); i.e., we can ask what connection is satisfied by int(x) and real(x). The answer is $\varphi=\subset^{-1}$.

26.3 Combined single quasi-relational databases

Combined single quasi-relational databases combine first and second-level single quasi-relational databases.

Example 26.3: Suppose we have a personal_information table shown in Table 26.12, and a is_subset_of(P, Q) table shown in Table 26.13.

Table 26.12 Personal_information

name	occupation	work_at	work_year	father
Little Wang	journalist	Beijing Daily	25	Big Wang

Table 26.13 Is_subset_of (P, Q)

P	Q
journalist	capable_of_writing
capable_of_writing	literate
literate	

Based on Table 26.12, we can establish binary views:

CREATE VIEW occupation_view(name, occupation)

AS

SELECT name, occupation
FROM personal_information

and

CREATE VIEW father_view(name, father)
AS
SELECT name, father
FROM personal_information

We have the following first-level Datalog statements:

ancestor(x, y): -father_view(x, y).
ancestor(x, z): -father_view(x, y), ancestor(y, z).

We have the following combined Datalog statements:

is_member_of(x, Q): -occupation_view(x, P) AND
is_subset_of(P, Q).
is_member_of(x, Q): -is_member_of(x, P) AND
is_subset_of(P, Q).

These two statements correspond to $\{x\in P\}\wedge\{P\subseteq^{-1}Q\}\subseteq^{-1}\{x\in Q\}$, where $x\in P$ and $x\in Q$ are facts, $P\subseteq^{-1}Q$ is an empirical or mathematical connection. Hence they are combined Datalog statements. We have the following second-level Datalog statement:

is_subset_of(P, R): -is_subset_of(P, Q) AND
is_subset_of(Q, R).

The statement corresponds to $\{P\subseteq^{-1}Q\}\cap\{Q\subseteq^{-1}R\}\subseteq^{-1}\{P\subseteq^{-1}R\}$.

The system can make the following first-level inference:

ancestor(x, y): -fataher_view(x, y) AND
x=Little Wang AND
y=Big Wang,

inferring that Big Wang is Little Wang's ancestor.

The system can make the following combined inference:

is_member_of(x, Q): -occupaation_view(x, P) AND
is_subset_of(P, Q) AND
x=Little Wang AND
P=journalist AND
Q=capable-of_writing,

inferring that Little Wang is capable of writing.

The system can make the following second-level inference:

is_subset_of(P, R): -is_subset_of(P, Q) AND
is_subset_of(Q, R) AND
P=journalist AND

Q=capable_of_writing AND

R=literate,

inferring that journalists are all literate.

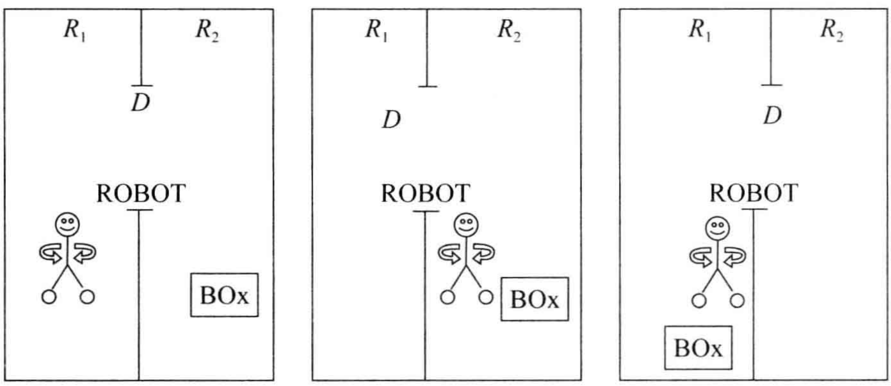
Combined single quasi_relational databases can be used to implement semantic networks: let the names of the binary views or binary tables be the tags of the directed arcs, let the first attributes be the starting vertices, let the second attributes be the arriving vertices. Such relations of semantic networks as is_subset_of, is_member_of, and is_part_of can all be implemented by combined single quasi-relational databases.

Chapter 27 Mutually-inversistic planning and scheduling

Mutually-inversistic planning and scheduling are based on second-level single quasi-Prolog.

27.1 Mutually-inversistic agent planning

Example 27.1: Consider the plan that a robot gets a box from the neighboring room. The three states are shown in Fig. 27.1.



(a) Initial state (b) Intermediate state (c) Goal state

Fig. 27.1 States of a robot catching a box

The predicates describing the states are as follows:

IN(rbt, r_l): robot rbt is in room r_l.

IN(b, r_l): box b is in room r_l.

CN(d, r_l, r_2): door d connects rooms r_l and r_2.

We use the lower case strings rbt, b, d, r_l, and r_2 to denote term variables, use upper case strings ROBOT, BOX, D, R_1, and R_2 to denote term constants the robot, the box, the door, room one and room two respectively.

The initial state is as follows:

IN(ROBOT, R_1)$\wedge$IN(BOX, R_2)$\wedge$CN(D, R_1, R_2). (27.1)

The goal state is as follows:

IN(ROBOT, R_1)$\wedge$IN(BOX, R_1)$\wedge$CN(D, R_1, R_2). (27.2)

Let the initial state mutually inversely implies the goal state, obtaining the initial state-goal state propposition:

$$\text{IN(ROBOT, R}_1) \wedge \text{IN(BOX, R}_2) \wedge \text{CN(D, R}_1\text{, R}_2) \leqslant^{-1} \text{IN(ROBOT, R}_1) \wedge \text{IN(BOX, R}_1) \wedge \text{CN(D, R}_1\text{, R}_2). \tag{27.3}$$

The initial state-goal state propositoin is a zeroth-order single empirical or mathematical connection proposition.

The robot has two actions: go through (gothru) and push through (pushthru).

Gothru(d, r_1, r_2): the robot goes from room r_1 through door d into room r_2.

Prestate: IN(rbt, r_1)$\wedge$IN(b, r_2)$\wedge$CN(d, r_1, r_2).

Poststate: IN(rbt, r_2)$\wedge$IN(b, r_2)$\wedge$CN(d, r_1, r_2).

Pushthru(b, d, r_2, r_1): the robot pushes box b from room r_2 through door d into room r_1.

Prestate: IN(rbt, r_2)$\wedge$IN(b, r_2)$\wedge$CN(d, r_1, r_2).

Poststate: IN(rbt, r_1)$\wedge$IN(b, r_1)$\wedge$CN(d, r_1, r_2).

Actions and their prestates, poststates form prestate-action-poststate axioms. The two axioms are as follows:

$$\{\text{IN}(rbt, r_1) \wedge \text{IN}(b, r_2) \wedge \text{CN}(d, r_1, r_2)\}\text{gothru}(d, r_1, r_2) \{\text{IN}(rbt, r_2) \wedge \text{IN}(b, r_2) \wedge \text{CN}(d, r_1, r_2)\} \tag{27.4}$$

$$\{\text{IN}(rbt, r_2) \wedge \text{IN}(b, r_2) \wedge \text{CN}(d, r_1, r_2)\}\text{pushthru}(b, d, r_2, r_1) \{\text{IN}(rbt, r_1) \wedge \text{IN}(b, r_1) \wedge \text{CN}(d, r_1, r_2)\} \tag{27.5}$$

These axioms are first-order single empirical or mathematical connection propositions.

The second-level single quasi-Prolog program and goal are as follows:

(1) {IN(rbt, r_2)$\wedge$IN(b, r_2)$\wedge$CN(d, r_1, r_2)}gothru(d, r_1, r_2){IN(rbt, r_1)$\wedge$IN(b, r_2) $\wedge$CN(d, r_1, r_2)}.

(2) {IN(rbt, r_1)$\wedge$IN(b, r_1)$\wedge$CN(d, r_1, r_2)}pushthru(b, d, r_2, r_1){IN(rbt, r_2)$\wedge$IN(b, r_2) $\wedge$CN(d, r_1, r_2)}.

(3) {R==>>P}===>>>{R>>P}.

(4) {R==>>P}===>>>{Q>>P}, {R==>>Q}.

(5) ?-{IN(ROBOT, R_1)$\wedge$IN(BOX, R_1)$\wedge$CN(D, R_1, R_2)}==>>{IN(ROBOT, R_1) $\wedge$IN(BOX, R_2)$\wedge$CN(D, R_1, R_2)}.

The two prestate-action-poststate axioms are the second-level unconditional clauses, the initial state-goal state proposition is the goal. The process of proving the initial state-goal state proposition to be a theorem is the process of generating the plan. In the process, the action sequence binding to >> is the plan generated: gothru(D, R_1, R_2), pushthru(BOX, D, R_2, R_1). Although the two axioms are the prestate-action-poststate axioms, they can be reduced to precondition-action-effect axioms. The second-level single quasi-Prolog program and goal are the second-level tail recursion with exit, which can be transformed to second-level iteration.

27.2 Mutually-inversistic multiagent planning

27.2.1 An example

Example 27.2: There are two manipulators: agent m and agent n. They can unstack or stack building blocks. There are four building blocks of the same size, marked 1, 2, 3, and 4.There are three tables, marked A, B, and C. The size of a table is just for one building block, on which can be put another building block. The position for the lower level building block is marked by L, the position for the upper level building block is marked by U. The initial state is shown in Fig. 27.2. The goal state for agent m is shown in Fig. 27.3, that for agent n is shown in Fig. 27.4.

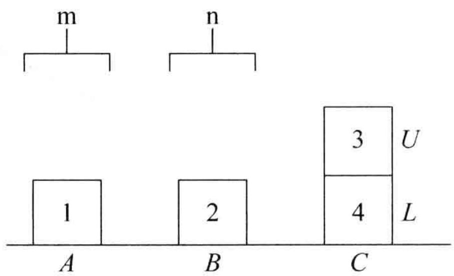


Fig. 27.2 Initial state

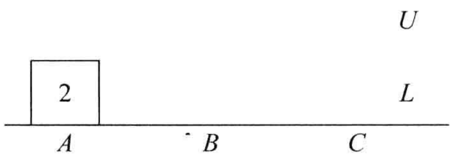


Fig. 27.3 Goal state for agent m

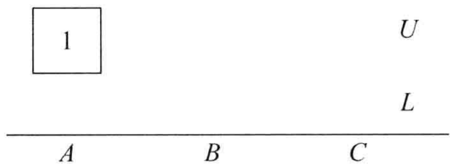


Fig. 27.4 Goal state for agent n

Agent m can do the following actions:

m-unstack(x, y, U): m unstack building block x from the U position of table y.

m-unstack(x, y, L): m unstack building block x from the L position of table y.

m-stack(x, y, U): m stack building block x on the U position of table y.

m-stack(x, y, L): m stack building block x on the L position of table y.

Agent n can do the similar actions.

The predicates describing states are as follows:

At(x, y, U): building block x is at the U position of table y.

At(x, y, L): building block x is at the L position of table y.

m-holding(x): m is holding building block x.

m-idle: m is idle.

n-holding(x): n is holding building block x.

n-idle: n is idle.

The actions of agent m and their preconditions and effects are described as follows (we adopt the closed-world assumption):

m-unstack(x, y, U)

Precondition: at(x, y, U), at(v, y, L), m-idle.

Effect: at(v, y, L), m-holding(x) (According to the closed world assumption, in the effect, there is no predicate at(x, y, U), which means the predicate is false).

m-unstack(x, y, L)

Precondition: at(x, y, L), m-idle.

Effect: m-holding(x).

m-stack(x, y, U)

Precondition: at(v, y, L), m-holding(x).

Effect: at(x, y, U), at(v, y, L), m-idle.

m-stack(x, y, L)

Precondition: m-holding(x).

Effect: at(x, y, L), m-idle.

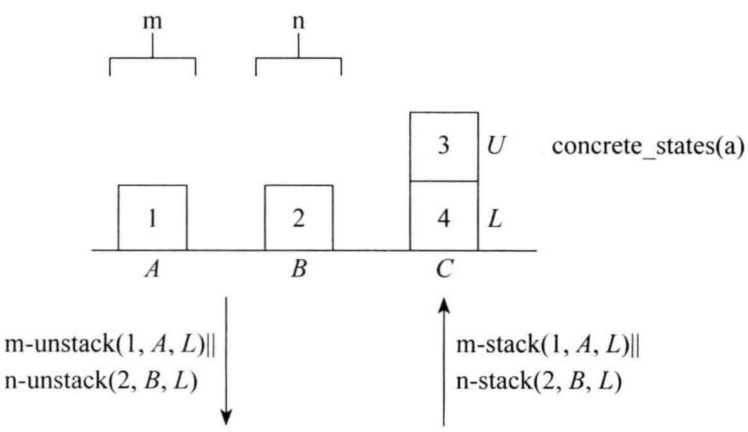
Agent n has similar actions and preconditions and effects.

Agent m's action m-unstack(x_1, y_a, L) and agent n's action n-unstack(x_2, y_b, L) can be combined to execute in parallel. We use m-unstack(x_1, y_a, L)||n-unstack(x_2, y_b, L) to denote their action combination, the prestate-action combination-poststate axiom of which is as follows:

$$\{\text{at}(x_1, y_a, \text{L})\wedge\text{at}(x_2, y_b, \text{L})\wedge\text{at}(x_3, y_c, \text{U})\wedge\text{at}(x_4, y_c, \text{L})\wedge\text{m-idle}\wedge\text{n-idle}\}\ \text{m-unstack}(x_1, y_a, \text{L})\|\text{n-unstack}(x_2, y_b, \text{L})\{\text{at}(x_3, y_c, \text{U})\wedge\text{at}(x_4, y_c, \text{L})\wedge\text{m-holding}(x_1)\wedge\text{n-holding}(x_2)\} \quad (27.6)$$

Similarly, other prestate-action combination-poststate axioms can be made. They are first-order single empirical or mathematical connection propositions.

Parts of the concrete states and their transitions are shown in Fig. 27.5. Parts of the abstract states and their transitions are shown in Fig. 27.6.



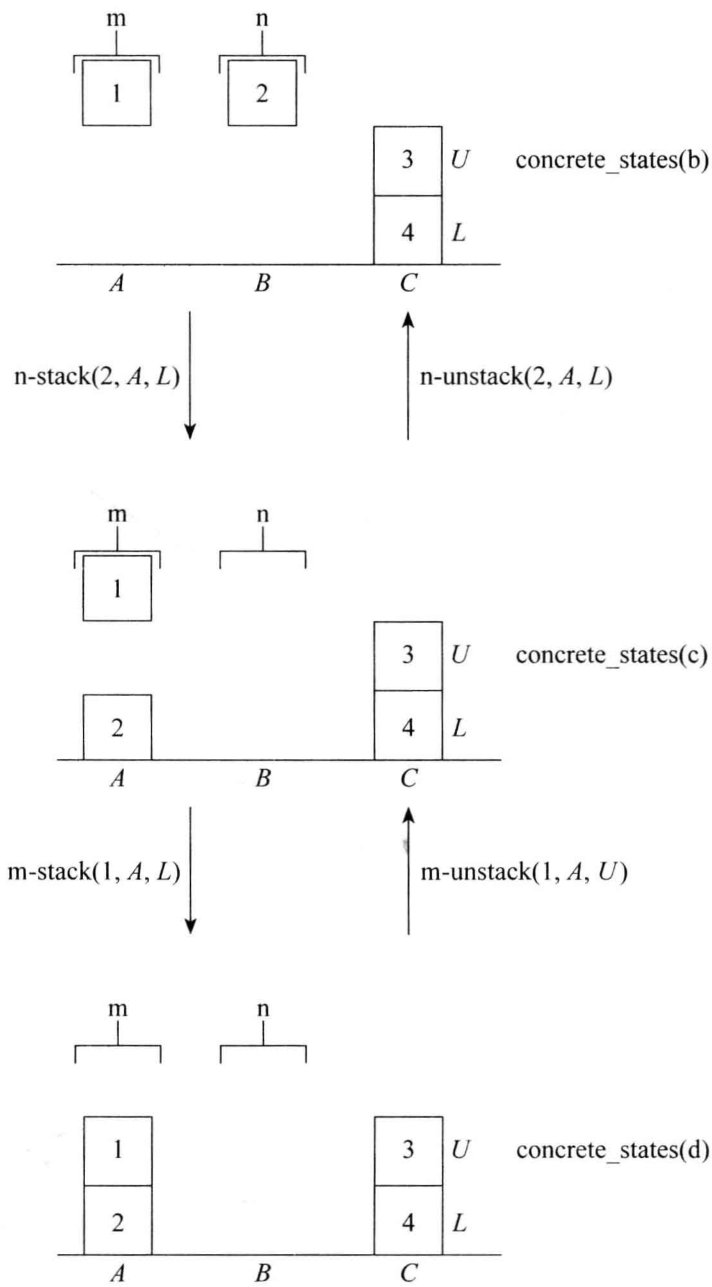


Fig. 27.5 Concrete states and their transitions

The initial state is described as follows:

at(1, A, L)$\wedge$at(2, B, L)$\wedge$at(3, C, U)$\wedge$at(4, C, L)$\wedge$m-idle$\wedge$n-idle (27.7)

The goal state of m (Fig. 27.3) and that of n (Fig. 27.4) do not conflict, they can form a goal states combination, which is not unique. One scheme is shown in Fig. 27.5 (d), it is described as follows:

at(1, A, U)$\wedge$at(2, A, L)$\wedge$at(3, C, U)$\wedge$at(4, C, L)$\wedge$m-idle$\wedge$n-idle (27.8)

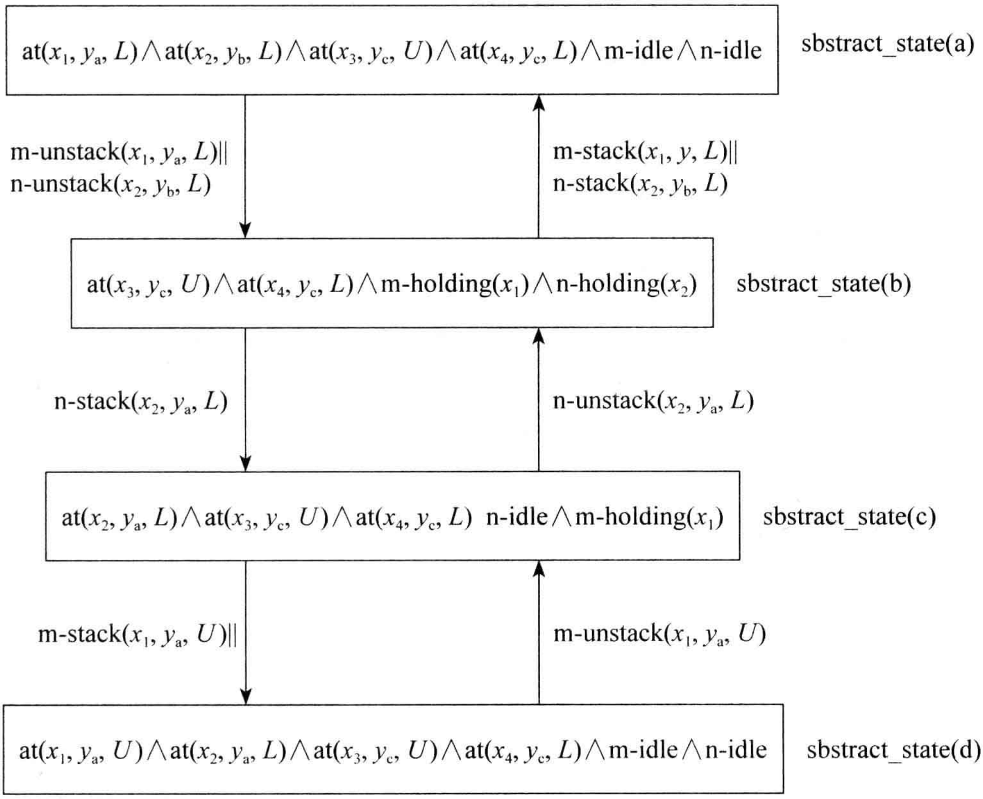


Fig. 27.6 Abstract states and their transitions

Let (27.7) mutually inversely implies (27.8), obtaining the initial state-goal states combination proposition, which is a zeroth-order single empirical or mathematical connection proposition:

{at(1, A, L)∧at(2, B, L)∧at(3, C, U)∧at(4, C, L)∧m-idle∧n-idle}$\leq^{-1}${at(1, A, U)∧at(2, A, L)∧at(3, C, U)∧at(4, C, L)∧m-idle∧n-idle} (27.9)

We aim at proving (27.9) to be a theorem. The proof process is one that generate the multiagent plan. The second-level single quasi-Prolog program and goal are as follows:

(1) {abstract_state(b)}m-unstack(x_1, y_a, L)||n-unstack(x_2, y_b, L){abstract_state(a)}.

(2) {abstract_state(c)}n-stack(x_2, y_a, L){abstract_state(b)}.

(3) {abstract_state(b)}m-stack(x_1, y_a, U){abstract_state(a)}.

(4) {*R*==>>*P*}===>>>{*R*>>*P*}.

(5) {*R*==>>*P*}===>>>{*Q*>>*P*}, {*R*==>>*Q*}.

(6) ?-{concrete_state(d)}==>>{concrete_state(a)}.

In the program and goal, clause (1) is the abbreviation of (27.6), clause (6) is the abbreviation of (27.9). The process of proving the initial state-goal states combination proposition to be a theorem is the process of generating the plan. The action sequence binding to >> is the plan generated: m-unstack(1, A, L)||n-unstack(2, B, L), n-stack(2, A,

L), m-stack(1, A, U), which is described by the transition from the concrete state (a) to the concrete state (d) in Fig. 27.5. The program and goal are the second-level tail recursion with exit, which can be transformed into second-level iteration. The prestate-action combination-poststate axioms can be reduced to precondition-action combination-effect axioms.

27.2.2 Transformation from linear logic theorem prover to mutually-inversistic multiagent planner

Literature (Rao, Kungas, Matskin, 2004) gives a linear logic theorem prover based Web service composition system of skiing. It has several inference rules including hypothetical syllogism. It has five axioms:

|-Skill$\rightarrow_{\text{SelectModel}}$Model

|-Height, Weight$\rightarrow_{\text{SelectLength}}$Length

|-Model, Length$\rightarrow_{\text{SelectSki}}$ProductNr, Sportshop

|-ProductNr, Shop$\rightarrow_{\text{GetPrice}}$Price

|-SportShop$\rightarrow$Shop

The first four axioms are Web services, the last one is an ontology. The theorem to be proved is:

|-Skill, Height, Weight$\rightarrow$Price

The system is a second-level inference rule system, because it has many inference rules including hypothetical syllogism. It can be transformed into second-level single quasi-Prolog by transforming the inference rules into single logical theorems, taking the axioms as single empirical or mathematical theorems, taking the second-level negative expression of hypothetical inference, second-level disjunctive inference, and the rule of change from $\vee$ to $\vee/^{-1}$ as the inference rules. The second-level single quasi-Prolog program and goal are as follows:

(1) {Model, Length}SelectModel||SelectLength{Skill, Height, Weight}

(2) {ProductNr, SportShop}SelectSki{Model, Length}.

(3) {Shop}Ontology{SportShop}.

(4) {Price}GetPrice{ProductNr, Shop}.

(5) $\{R\}$==>>$\{P\}$===>>>$\{Q\}$==$\{P\}$, $\{R\}$==>>$\{Q\}$.

(6) $\{S\}$==>>$\{P, R\}$===>>>$\{Q\}$==$\{R\}$, $\{S\}$==$\{P, Q\}$.

(7) ?-{Price}==>>{Skill, Height, Weight}.

The system is a mutually-inversistic multiagent planner. The actions are in the middle of (1) through (4) with the preconditions following them and the effects in front of them. The proof process is one that generates the plan. In the process, the action sequence binding to == is the plan generated: SelectModel||SelectLength, SelectSki, Ontology, GetPrice.

27.3 Mutually-inversistic multiagent scheduling

Example 27.3: A truck (agent A) and a vessel (agent B) carry soldiers from the rear to the front. Agent A can carry 40 soldiers, goes forward and backward each need 30 hours. Agent B can carry 100 soldiers, when going forward, it is with the stream, and need 50 hours; when going backward, it is against the current, and need 70 hours. The scheduling diagram is shown in Fig. 27.7.

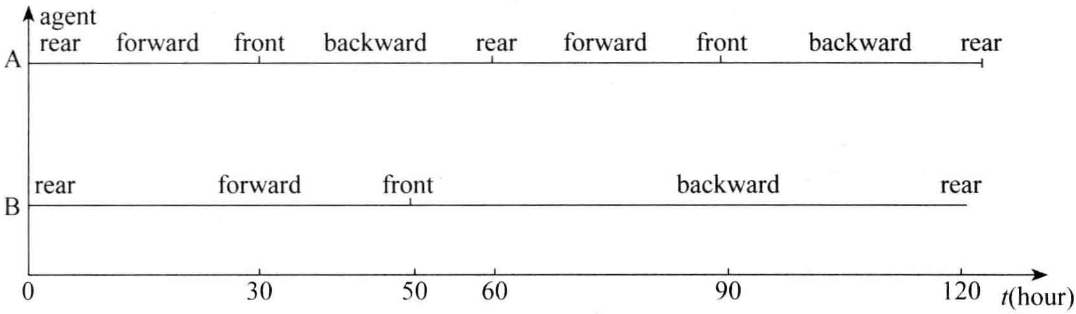


Fig. 27.7 Scheduling diagram

The time A reaches the front or the rear is called the turning point of A, because at this time A needs to turn directions. Likewise, we can define the turning point of B. At least one of the turning point of A or B is called the switch point. In Fig. 27.7, there are six switch points: 0, 30 hours, 50 hours, 60 hours, 90 hours, 120 hours. Take 30 hours as an example, at this time, the scheduling scheme is from A going forward and B going forward switched to A going backward and B going forward.

We use t_{Af} to denote the time A has gone forward at the switch point, use t_{Ab} to denote the time A has gone backward at the switch point, use D_A to denote more time A needs to go to the next turning point. T_{Bf}, t_{Bb}, D_B can be similarly defined. We use t_i to denote the time interval between two switch points, use t to denote the total scheduling time, use *sld* to denote the number of soldiers reaching the front.

A has two actions. Forward$_A$(sld$_1$, t_i) denotes that A goes forward t_i time, carries as much as sld_l soldiers. Backward$_A$(t_i) denotes that A goes backward t_i time. Forward$_B$(sld_2, t_i) and backward$_B$(t_i) can be similarly defined.

There are four action combinations: A going forward and B going forward, A going backward and B going forward, A going forward and B going backward, and A going backward and B going backward. Each action combination can be described by a prestate-action combination-poststate axiom, whose prestate and poststate are both switch points, and who is a first-order single empirical or mathematical connection proposition. The prestate-action combination-poststate axiom for the action combination A going forward and B going

forward is given in (27.10) where the prestate is in front of the action combination and the poststate follows it.

$$\{t_{Af}<30\wedge t_{Bf}<50\wedge t=\mathrm{c}\wedge sld=\mathrm{d}\wedge t_{Ab}=30\wedge t_{Bb}=70\}\mathrm{forward}_{\mathrm{A}}(40,\ t_i)||\mathrm{forward}_{\mathrm{B}}(100,\ t_i)$$
$$\{(D_A=30-t_{Af})\wedge(D_B=50-t_{Bf})[\mathrm{if}(D_A=D_B)\ \mathrm{then}\ (t_i=D_A=D_B\wedge t_{Ab}=0\wedge t_{Bb}=0\wedge(sld=\mathrm{d}+140))\ \mathrm{else\ if}\ (D_A<D_B)\ \mathrm{then}\ (t_i=D_A\wedge t_{Ab}=0\wedge(sld=\mathrm{d}+40))\ \mathrm{else}\ (t_i=D_B\wedge t_{Bb}=0\wedge(sld=\mathrm{d}+100))]\wedge(t_{Af}=t_{Af}+t_i)\wedge(t_{Bf}=t_{Bf}+t_i)\wedge(t=\mathrm{c}+t_i)\} \tag{27.10}$$

The initial state is as follows:

$$t_{Af}=0\wedge t_{Bf}=0\wedge t=0\wedge sld=0\wedge t_{Ab}=30\wedge t_{Bb}=70 \tag{27.11}$$

The goal state is as follows:

$$t_{Ab}<30\wedge t_{Bb}<70\wedge t=240\wedge sld\geqslant 350\wedge t_{Af}=30\wedge t_{Bf}=50 \tag{27.12}$$

Let the initial state mutually inversely implying the goal state be the initial state-goal state proposition, which is a zeroth-order single empirical or mathematical connection proposition.

Mutually-inversistic multiagent scheduler is second-level single quasi-Prolog. The second-level unconditional clauses are the prestate-action combination-poststate axioms, the second-level goal is the initial state-goal state proposition to be proved, the second-level conditional clauses are the same as clauses (4) and (5) of Example 27.2.

In certain step, (27.11) unifies with the prestate of (27.10) successfully, and the action combination $\mathrm{forward}_{\mathrm{A}}(40,\ t_i)||\mathrm{forward}_{\mathrm{B}}(100,\ t_i)$ is executed. Then the poststate is computed: $D_A=30-t_{Af}=30-0=30$, $D_B=50-t_{Bf}=50-0=50$; $D_A<D_B$, hence, $t_i=D_A=30$, $t_{Ab}=0$, $sld=0+40=40$, $t_{Af}=0+30=30$, $t_{Bf}=0+30=30$, $t=0+30=30$, t_{Bb} remains 70 unchanged. The next action combination is A going backward and B going forward.

The advantage of the mutually-inversistic multiagent scheduler is that it solves the resource representation problem which other schedulers do not solve. The disadvantage of it is that its number of prestate-action combination-poststate axioms grows exponentially with the number of agents n.

Chapter 28
Mutually-Inversistic Semantic Network

Mutually-inversistic semantic network is an inference semantic network. It is divided into first-level semantic network and second-lelvel semantic network, based on the first-level predicate calculus and second-level predicate calculus respectively. Traditional semantic network except partitioned semantic network is part of first-level semantic network.

28.1 First-level semantic network

First-level semantic network is used to describe the second-order main and auxiliary; i.e., predicates and functions, and the third-order main and auxiliary; i.e., empirical or mathematical connection operators and fact composition operators.

28.1.1 Predicates and functions

It is convenient to denote a binary predicate by the first-level semantic network, in which the binary predicate is denoted by a directed arc connecting two vertices, the directed arc represents the predicate, the vertex the arc leaves represents the first argument of the predicate, the vertex the arc enters represents the second argument of the predicate. For example, the binary predicate parent(John, Sam) denoted by first-level semantic network is shown in Fig. 28.1.

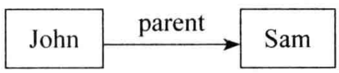


Fig. 28.1 Binary predicate

It is also convenient to denote a unary predicate by the first-level semantic network. For example, the unary predicate shine(Sun) can be denoted by Fig. 28.2.

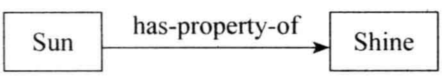


Fig. 28.2 Unary predicate

A ternary predicate can be transformed into four binary predicates. For example, the ternary predicate give(John, Mary, book) can be transformed into Isa(G1, giving_

event)$\wedge$giver(G_1, John)$\wedge$recip(G_1, Mary)$\wedge$obj(G_1, book). Its corresponding first-level semantic network is shown in Fig. 28.3.

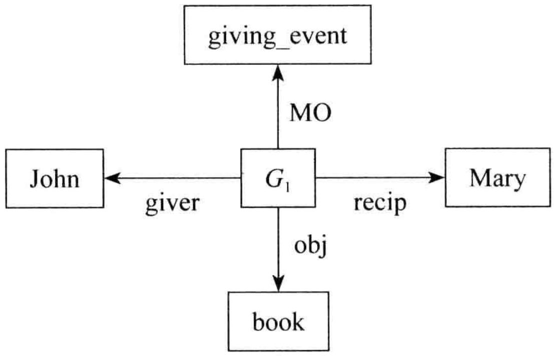


Fig. 28.3 Ternary predicate

Likewise, a quadrinary predicate can be transformed into five binary predicates.

A unary function can be denoted like a binary predicate. For example, the unary function height(Zhou)=1.76m can be denoted by Fig. 28.4.

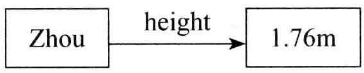


Fig. 28.4 Unary function

A binary function can be denoted like a ternary predicate. For example, the binary function 3+2=5 can be denoted by Fig. 28.5.

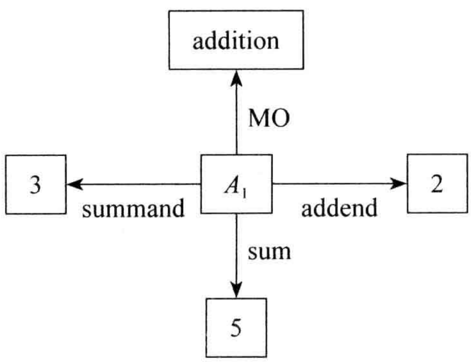


Fig. 28.5 Binary function

28.1.2 Empirical or mathematical connection operators and fact composition operators

There are two ways of denoting empirical or mathematical connection operators and fact composition operators. The first way is that empirical or mathematical connection

operators are denoted by arcs, fact composition operators are denoted by vertices. The second way is that both empirical or mathematical connection operators and fact composition operators are denoted by closed lines.

28.1.2.1 The first way of denoting empirical or mathematical connection operators and fact composition operators

There are two cases to be considered. The first case is that the antecedent of the empirical or mathematical connection operator has only one fact proposition, and the fact composition operator is not needed. The second case is that the antecedent of the empirical or mathematical connection operator has two or more fact propositions, and the fact composition operator is needed.

28.1.2.1.1 The first case

In this case, examples of using directed arcs to denote $\leqslant^{-1}$ are given in Figs. 28.6 and 28.7.

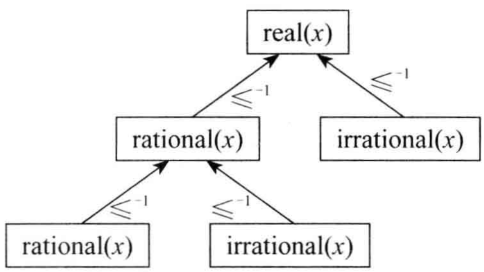


Fig. 28.6 Using directed arcs to denote $\leqslant^{-1}$

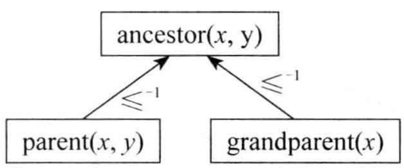


Fig. 28.7 Using directed arcs to denote $\leqslant^{-1}$

An example of using directed arc to denote $/\wedge^{-1}$ is given in Fig. 28.8.

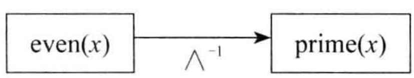


Fig.28.8 Using directed arc to denote $/\wedge^{-1}$

When the fact propositions are unary predicates, then the term variables in the unary predicates can be omitted. For example, in the bronze classification semantic network, bronze(x) can be written as bronze. The semantic network is shown in Fig. 28.9.

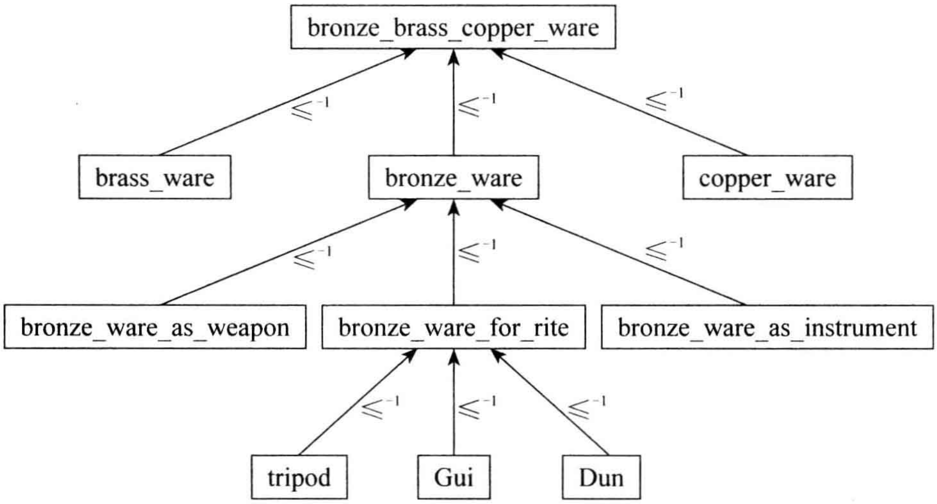


Fig. 28.9 Term variables can be omitted

When $\leqslant^{-1}$ is denoted by a directed arc, $\leqslant^{-1}$ corresponds to KO (KIND_OF). The other relations used in the mutually-inversistic semantic network include PO (PART_OF), MO (MUMBER_OF), IO (INSTANCE_OF), AO (ATTRIBUTE_OF), SA (SAME_AS), CO (CONJUNCT_OF), DO (DISJUNCT_OF), NO (NEGATION_OF).

28.1.2.1.2 The second case

In this case, we need vertices to denote fact composition operators, see the family semantic network shown in Fig. 28.10.

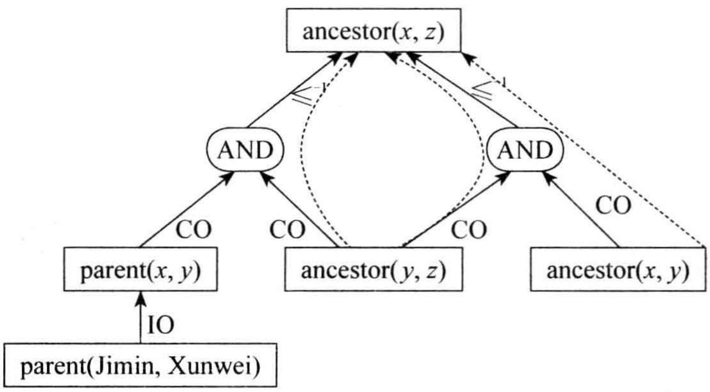


Fig. 28.10 Using vertices to denote fact composition operators

Fig. 28.10 can be used to discover recursions. If we start from certain predicate, say ancestor, via an upward route, and reach the same predicate, then a recursion exists. The dotted lines in Fig. 28.10 denote recursion.

A part of the first-level semantic network for animal classification expert system is shown in Fig. 28.11.

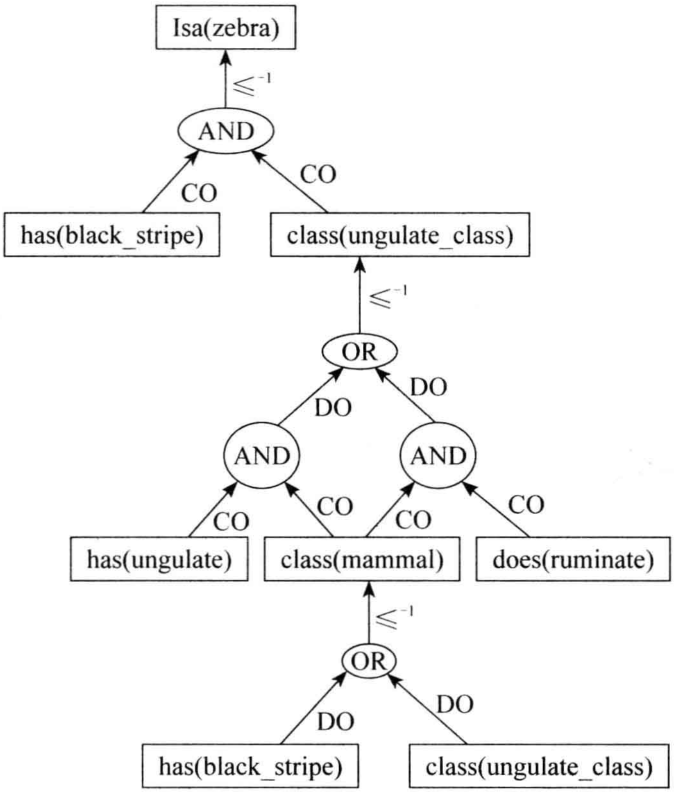


Fig. 28.11 Animal classification expert system

28.1.2.2 The second way of denoting empirical or mathematical connection operators and fact composition operators

In this way, we use directed arcs to denote predicates, use vertices to denote terms, use closed dotted lines marked with ANTE, CONSE to denote the antecedent, and consequent of the empirical or mathematical connection operator $\leqslant^{-1}$ respectively, use closed dotted lines marked with CONJ, DISJ to denote conjunction, and disjunction respectively.

For example, the first-level semantic network of parent(John, Sam)$\leqslant^{-1}$ancestor(John, Sam) is shown in Fig. 28.12.

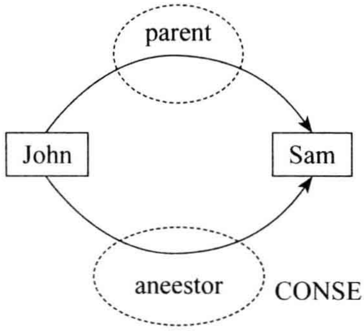


Fig. 28.12 The second way of denoting empirical or mathematical connection operators and fact composition operators

Another example is that the first-level semantic network of parent(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z) is shown in Fig. 28.13.

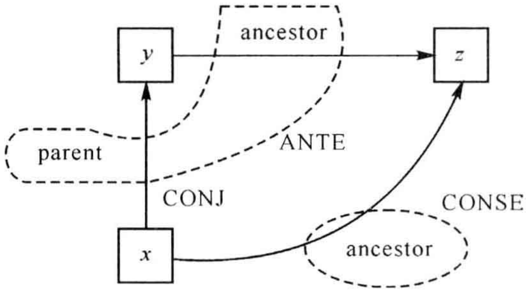


Fig. 28.13 The second way of denoting empirical or mathematical connection operators and fact composition operators

From Fig. 28.13, we see the second way of denoting the fact composition operator$\wedge$ (CONJ): using CONJ to circle two or more predicates to denote the conjunction of these predicates.

28.1.3 First-level semantic network representation of quantifier-free propositions

In the previous section, we used the first-level semantic network to denote the quantifier-free first-order single empirical or mathematical connection propositions. Parts of Fig. 28.6, Fig. 28.8, and Fig. 28.13 denote first-order single empirical or mathematical connection propositions integer(x)$\leqslant^{-1}$rational(x), even_number(x)/$\wedge^{-1}$prime_number(x), and parent(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z) respectively. They correspond to propositions with quantifier in classical logic: $\forall_x$(integer(x)$\rightarrow$rational(x)), $\exists_x$(even_number(x)$\wedge$prime_number(x)), and $\forall_x\forall_y\forall_z$(parent($x$, y)$\wedge$ancestor(y, z)$\rightarrow$ancestor(x, z)) respectively. In this section, we use the first-level semantic network to denote the quantifier-free first-order multiple empirical or mathematical connection propositions, and compare them with the partitioned semantic network of propositions with quantifier in classical logic.

First, let us consider the first-order multiple empirical or mathematical connection propositions with the nonproperty propositions being the binary predicates. Suppose we have the statement: Every dog has bitten a postman. In classical logic, this statement is denoted by the proposition $\forall_x$(dog(x)$\rightarrow\exists_y$(postman(y)$\wedge$bite(x, y))). The partitioned semantic network of the proposition is shown in Fig. 28.14.

In mutually-inversistic logic, the statement is denoted by the proposition dog(x)$\leqslant^{-1}$postman(y)/$\wedge^{-1}$bite(x, y). The first-level semantic network of the proposition is shown in Fig. 28.15.

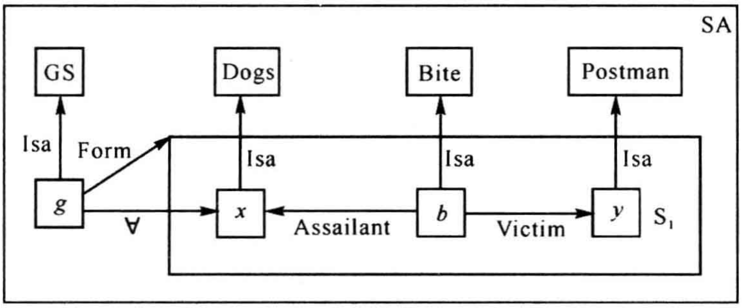


Fig. 28.14 Partitioned semantic network for every dog has bitten a postman

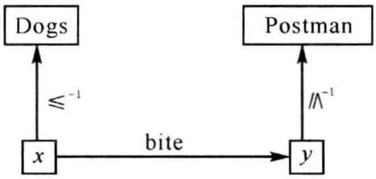


Fig. 28.15 First-level semantic network for every dog has bitten a postman

Suppose we have the statement: Every dog has bitten every postman. In classical logic, this statement is denoted by the proposition $\forall_x(\text{dog}(x)\rightarrow\forall_y(\text{postman}(y)\rightarrow\text{bite}(x, y)))$. The partitioned semantic network of the proposition is shown in Fig. 28.16.

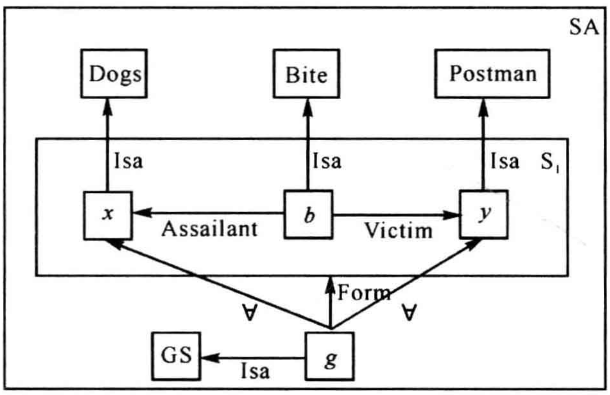


Fig. 28.16 Partitioned semantic network for every dog has bitten every postman

In mutually-inversistic logic, the statement is denoted by the proposition $\text{dog}(x)\leqslant^{-1}\text{postman}(y)\leqslant^{-1}\text{bite}(x, y)$. The first-level semantic network of the proposition is shown in Fig. 28.17.

Suppose we have the statement: Every dog in town has bitten the constable. In classical logic, the statement is denoted by the proposition $\exists_y(\text{constable}(y)\wedge\forall_x(\text{dog}(x)\rightarrow\text{bite}(x, y)))$. The partitioned semantic network of the proposition is shown in Fig. 28.18.

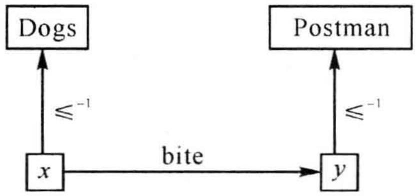


Fig. 28.17 First-level semantic network for every dog has bitten every postman

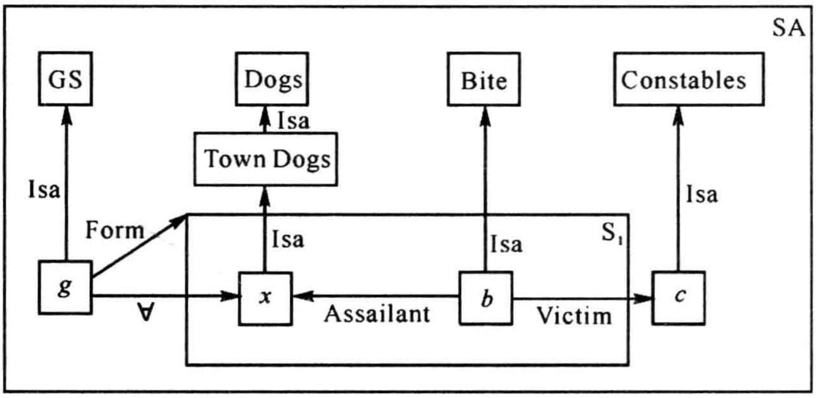


Fig. 28.18 Partitioned semantic network for every dog in town has bitten the constable

In mutually-inversistic logic, the statement is denoted by the proposition constable(John)$/\wedge^{-1}\{$dog$(x)\leqslant^{-1}$bite$(x$, John$)\}$. The first-level semantic network of the proposition is shown in Fig. 28.19.

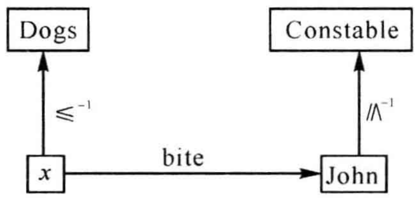


Fig. 28.19 First-level semantic network for every dog in town has bitten the constable

Comparing Figs. 28.15 with 28.14, 28.17 with 28.16, and 28.19 with 28.18, we see that the expressive power of first-level semantic network is the same as that of partitioned semantic network, but the former is simpler than the latter.

Then, let us consider the first-order multiple empirical or mathematical connection propositions with the nonproperty propositions being ternary predicates. Suppose we have the statement: every customer purchases some supplier's all commodities. In mutually-inversistic logic, the statement is denoted by the proposition customer$(x)\leqslant^{-1}$supplier$(y)/\wedge^{-1}$ $\{$commodity$(z)\leqslant^{-1}$purchase$(x, y, z)\}$. The first-level semantic network of the proposition is shown in Fig. 28.20. For the same statement, the partitioned semantic network representation is not reported.

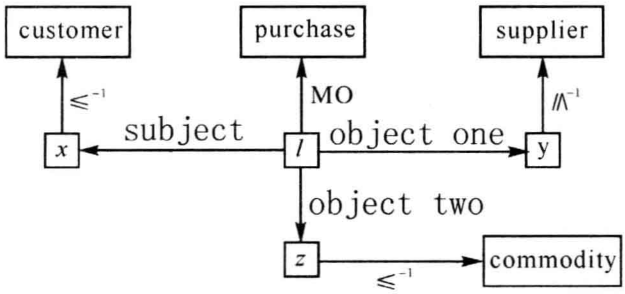


Fig. 28.20 First-level semantic network for nonproperty proposition being ternary predicate

28.1.4 Inferences of first-level semantic networks

First-level semantic network has inheritance inference and match inference, both are hypothetical inference.

28.1.4.1 Inheritance inferences

Example 28.1: Using inheritance inference to make Socrates inference.

Solution: Suppose we have a first-level semantic network shown in Fig. 28.21.

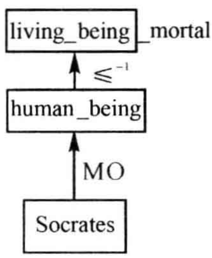


Fig. 28.21 Inheritance inference

The famous Socrates inference is: all men are mortal, Socrates is a man, therefore, Socrates is mortal. In Fig. 28.21, Socrates inference is made as follows:

human beings inherit the attribute of mortality from living beings, and Socrates inherits the attribute of mortality from human beings.

28.1.4.2 Match inferences

The fact network and goal network in traditional semantic network correspond to the network of the major premise and that of the minor premise-conclusion in mutually-inversistic semantic network including first-level semantic network and second-level semantic network.

Example 28.2: Make the following inference using match inference: from the major premise parent(x, y)$\wedge$ancestor(y, z)$\leqslant^{-1}$ancestor(x, z) and the minor premise parent(John,

Sam)$\wedge$ancestor(Sam, Bob) to infer the conclusion ancestor(John, Bob).

Solution: The network of the major premise has been shown in Fig. 28.13, the network of the minor premise-conclusion is shown in Fig. 28.22.

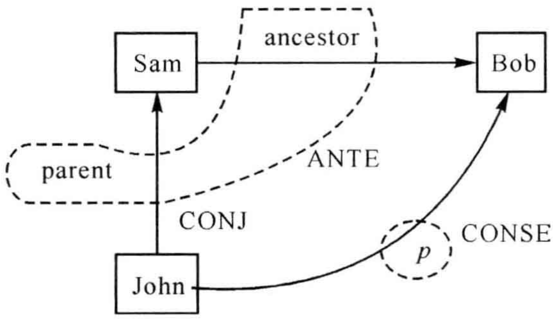


Fig. 28.22 Network of the minor premise-conclusion one of the match inference

Matching Figs. 28.13 and 28.22, we obtain the substitution {x/John, y/Sam, z/Bob, p/ ancestor}, and infer the conclusion ancestor(John, Bob).

Example 28.3: Make the following inference using match inference: from the major premise dog(x)$\leqslant^{-1}$postman(y)$/\wedge^{-1}$bite(x, y) and the minor premise dog(Huaner) to infer the conclusion postman(y)$/\wedge^{-1}$bite(Huaner, y).

Solution: The network of the major premise has been shown in Fig. 28.15, the network of the minor premise-conclusion is shown in Fig. 28.23.

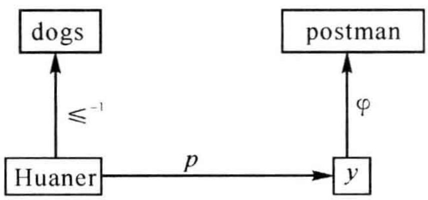


Fig. 28.23 Network of the minor premise-conclusion two of the match inference

Matching Figs. 28.15 and 28.23, we obtain the substitution {x/Huaner, p/bite, φ//$\wedge^{-1}$}, and infer the conclusion postman(y)$/\wedge^{-1}$bite(Huaner, y).

Example 28.4: Make the following inference using match inference: from the major premise parent(John, Sam) $\leqslant^{-1}$ancestor(John, Sam) and the minor premise parent(John, Sam) to infer the conclusion ancestor(John, Sam).

Solution: The network of the major premise has been shown in Fig. 28.12, the network of the minor premise-conclusion is shown in Fig. 28.24.

Matching Figs. 28.12 and 28.24, we obtain substitution {p/ancestor}, infer the conclusion ancestor(John, Sam).

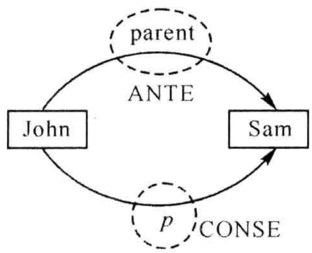


Fig. 28.24 Network of the minor premise-conclusion three of the match inference

28.2 Second-level semantic networks

28.2.1 Logical connection operators and empirical or mathematical composition operators

Similar to empirical or mathematical connection operators and fact composition operators, there are also two ways of denoting logical connection operators and empirical or mathematical composition operators. For the first way, there are also two cases. The first case is that the antecedent of the logical connection operator has only one single empirical or mathematical connection proposition. The second case is that the antecedent of the logical connection operator has two or more single empirical or mathematical connection propositions.

28.2.1.1 First way of denoting logical connection operators and empirical or mathematical composition operators

28.2.1.1.1 The first case

An example of this case is shown in Fig. 28.25.

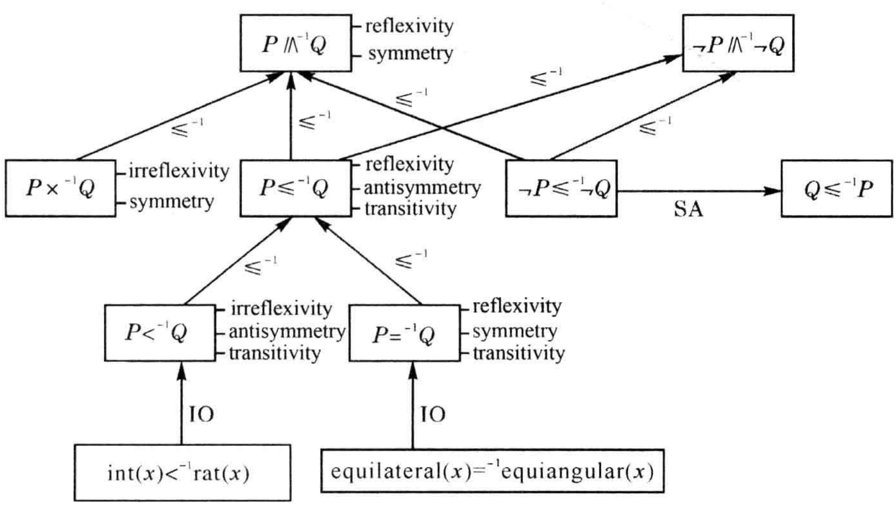


Fig. 28.25 The antecedent has only one empirical or mathematical connection proposition

Unlike the first-level semantic network, in Fig. 28.25, the descendant vertices do not inherit the attributes from the ancestor vertices. For example, vertex $P/\wedge^{-1}Q$ is reflexive, while its descendant vertex $P<^{-1}Q$ is not.

28.2.1.1.2 The second case

Examples of this case are shown in Figs. 28.26 and 28.27.

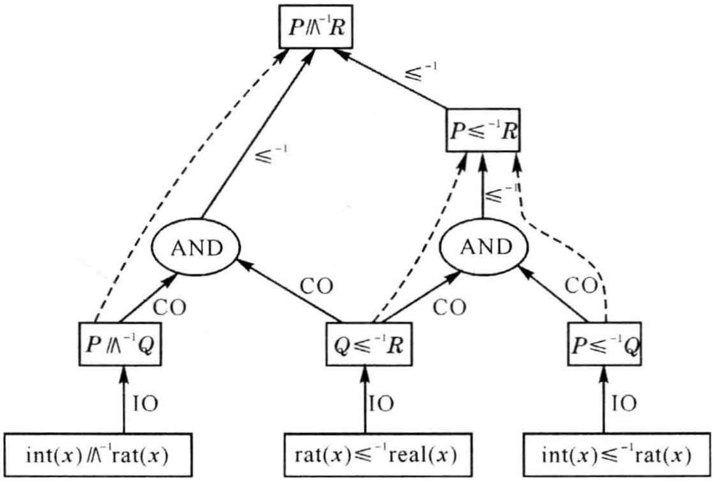


Fig. 28.26 The antecedent has two or more empirical or mathematical connection propositions

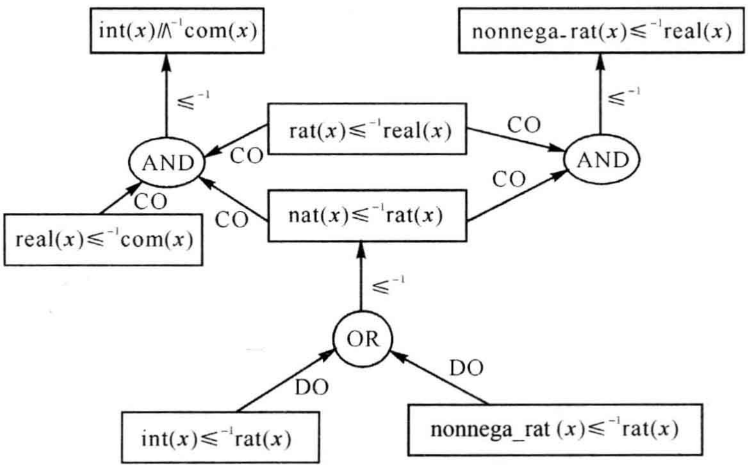


Fig. 28.27 Second-level single quasi-expert system of number system

Fig. 28.26 can be used to discover second-level recursion. If we start from certain empirical or mathematical connection operator, via an upward route, reach the same empirical or mathematical connection operator, then a second-level recursion exists, as is shown by the dotted lines in Fig. 28.26. Fig. 28.27 is the second-level single quasi-expert system of number system.

28.2.1.2 Second way of denoting logical connection operators and empirical or mathematical composition operators

In this way, the second-level semantic network of the second-order single logical connection proposition $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is shown in Fig. 28.28.

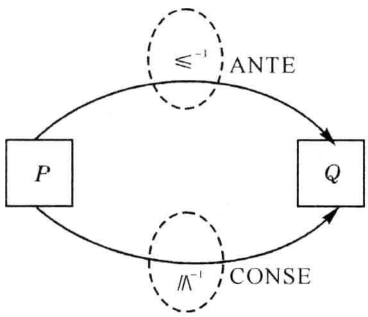


Fig. 28.28 Second way of denoting logical connection operators and empirical or mathematical composition operators

The second-level semantic network of the first-order single logical connection proposition $\{\text{tripod}(x)\leqslant^{-1}\text{bronzeware\_for\_rite}(x)\}\wedge\{\text{bronzeware\_for\_rite}(x)\leqslant^{-1}\text{bronzeware}(x)\}\wedge\{\text{bronzeware}(x)\leqslant^{-1}\text{bronze\_brass\_copper\_ware}(x)\}\leqslant^{-1}\{\text{tripod}(x)\leqslant^{-1}\text{bronze\_brass\_copper\_ware}(x)\}$ is shown in Fig. 28.29.

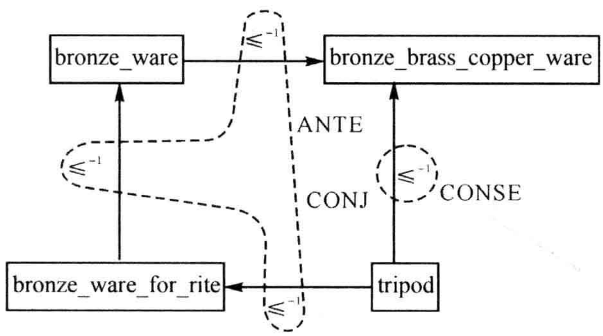


Fig. 28.29 Second way of denoting logical connection operators and empirical or mathematical composition operators

28.2.2 Inferences of second-level semantic networks

Because Fig.28.25 do not have inheritance, second-level semantic networks do not have inheritance inference, they only have match inference.

Example 28.5: Make the following inference using match inference: from the major premise $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ and the minor premise integer$(x)\leqslant^{-1}$rational(x) to infer the conclusion integer$(x)/\wedge^{-1}$rational(x).

Solution: The network of the major premise has been shown in Fig. 28.28, the network

of the minor premise-conclusion is shown in Fig. 28.30.

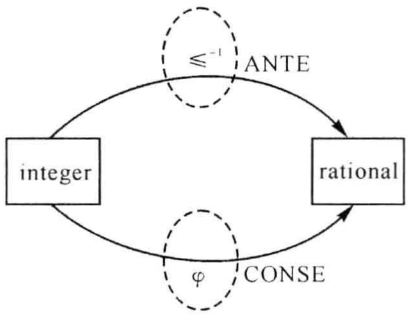


Fig. 28.30 Network of the minor premise-conclusion one of the match inference

Matching Figs. 28.28 with 28.30, we obtain the substitution {P/integer, Q/rational, φ//$\wedge^{-1}$}, infer the conclusion integer(x)/$\wedge^{-1}$rational(x).

Example 28.6: Make the following inference using match inference: from the major premise {tripod(x)$\leqslant^{-1}$bronzeware_for_rite(x)}$\wedge${bronzeware_for_rite(x)$\leqslant^{-1}$ bronzeware(x)}$\wedge${bronzeware(x)$\leqslant^{-1}$bronze_brass_copper_ware(x)}$\leqslant^{-1}${tripod(x)$\leqslant^{-1}$ bronze_brass_copper_ware(x)} and the minor premise {tripod(x)$\leqslant^{-1}$bronzeware_for_rite(x)}$\wedge${bronzeware_for_rite(x)$\leqslant^{-1}$bronzeware(x)}$\wedge${bronzeware(x)$\leqslant^{-1}$bronze_brass_copper_ware(x)} to infer the conclusion tripod(x)$\leqslant^{-1}$ bronze_brass_copper_ware(x).

Solution: The network of the major premise has been shown in Fig. 28.29, the network of the minor premise-conclusion is shown in Fig. 28.31.

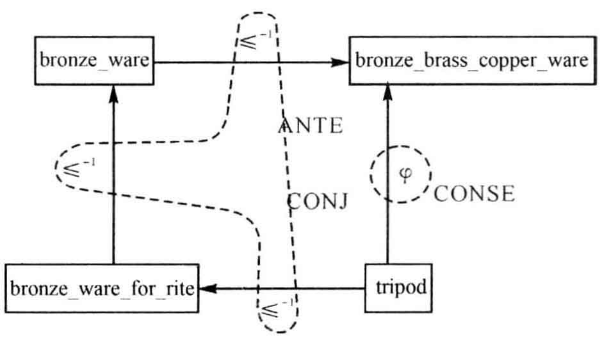


Fig. 28.31 Network of the minor premise-conclusion two of the match inference

Matching Figs. 28.29 with 28.31, we obtain the substitution {φ/$\leqslant^{-1}$}, infer the conclusion tripod(x)$\leqslant^{-1}$ bronze_brass_copper_ware(x).

28.3 Conclusions about semantic networks

We can make 10 conclusions about semantic networks:

(1) A first-level semantic network is quantifier-free, it has the same expressive power with a partitioned semantic network, but is much simpler.

(2) Mutually-inversistic semantic network has strong expressive power, it can express from the first-order to the fourth-order main and auxiliary.

(3) Second-level semantic network is first proposed by the author.

(4) Mutually-inversistic semantic network can denote expert systems. The first-level single quasi-expert system is shown in Fig. 28.11. The second-level single quasi-expert system is shown in Fig. 28.27.

(5) Mutually-inversistic semantic network is an inference semantic network, it can make hypothetical inference. The first-level hypothetical inferences are seen in Examples 28.1 through 28.4. The second-level hypothetical inferences are seen in Examples 28. 5 and 28.6.

(6) Mutually-inversistic semantic network can be used to carry out proofs of mutually-inversistic Prolog and expert systems. Examples of Prolog are seen in Examples 28.1 and 28.2. The example of first-level expert system is seen in Example 28.4. The example of second-level single quasi-Prolog is seen in Example 28.5. The example of second-level single quasi-expert system is seen in Example 28.6.

(7) Mutually-inversistic semantic network can be used to discover recursion and second-level recursion. Fig. 28.10 can be used to discover recursion. Fig. 28.26 can be used to discover second-level recursion.

(8) In the first way of denoting an empirical or mathematical connection proposition, an arc denotes an empirical or mathematical connection operator, in the second way, the arc denotes a predicate. In denoting a multiple empirical or mathematical connection proposition, some arcs denote the empirical or mathematical connection operators, others denote the predicates.

(9) Traditional semantic network is not stringent. While mutually-inversistic semantic network is stringent both on knowledge representation and on knowledge inference.

(10) If we establish knowledgebase on the basis of mutually-inversistic semantic network, then we can lift the knowledgebase from knowledge engineering to knowledge science.

Chapter 29
Mutually-inversistic expert systems

29.1 A semantic network

A first-level semantic network of bronze, brass, or copper ware is shown in Fig. 29.1.

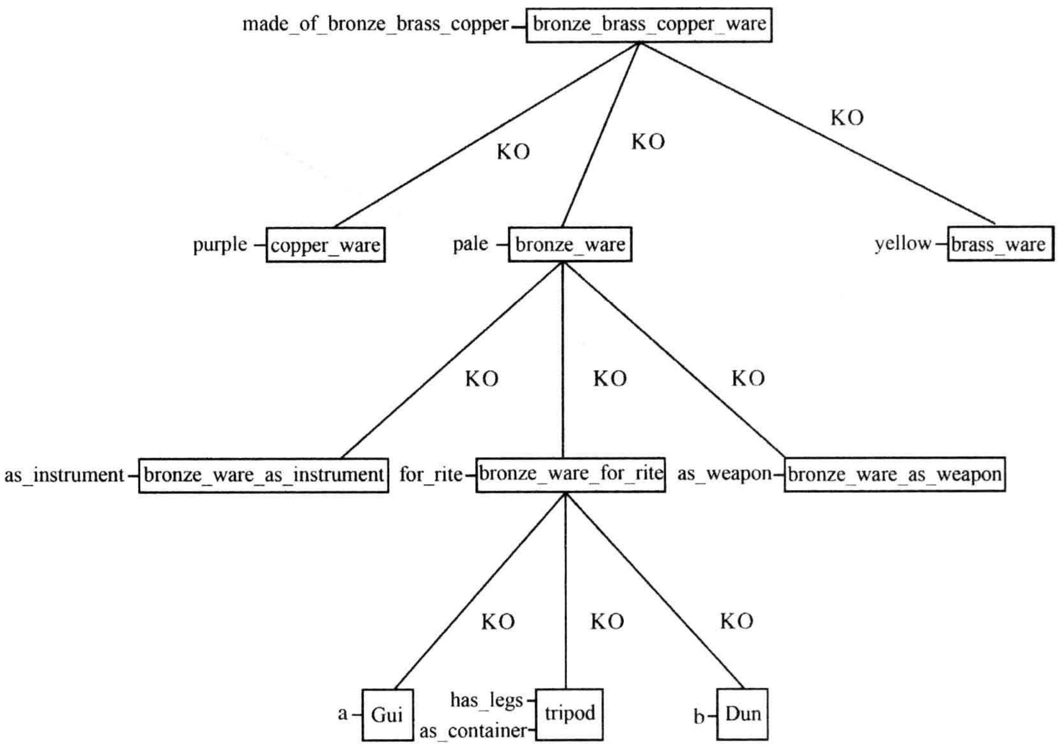


Fig. 29.1 First-level semantic network for bronze, brass, or copper ware

29.2 First-level single quasi-expert systems

Based on Fig. 29.1, we can construct the first-level single quasi-expert system, where each edge of Fig. 29.1 is a rule clause of Prolog, a zeroth-order single empirical or mathematical connection proposition. For example, the edge from bronze ware to bronze ware for rite can be denoted by:

class(bronze_ware_for_rite):-class(bronze_ware), attr(for_rite).

The Prolog program of the first-level single quasi_expert system for bronze, brass, or copper ware classification is as follows:

```
run: -isa(x),!.
     write(I think that it is a), writeln(x).
run: -writeln(I do not know what it is).
isa(gui): -class(bronze_ware_for_rite),
          attr(a).
isa(tripod): -class(bronze_ware_for_rite),
             attr(has_legs),
             attr(as_container).
isa(dun): -class(bronze_ware_for_rite),
          attr(b).
class(bronze_ware_as_instrument): -class(bronze_ware),
                                   attr(as_instrument).
class(bronze_ware_for_rite): -class(bronze_ware),
                              attr(for_rite).
class(bronze_ware_as_weapon): -class(bronze_ware),
                               attr(as_weapon).
class(copper_ware): -class(bronze_brass_copper_ware),
                     attr(purple).
class(bronze_ware): -class(bronze_brass_copper_ware),
                     attr(pale).
class(brass_ware): -class(bronze_brass_copper_ware),
                    attr(yellow).
attr(x): -write(Has it the attribute), write(x),write(?>>),
         read(Rep),
         same(Rep, y),
         assert(attr(x)).
         same(x, x).
```

29.3 Bottom-up second-level single quasi_expert systems

Each edge of Fig. 29.1 bottom-up is a first-order single empirical or mathematical connection proposition. For example, the edge from bronze ware to bronze, brass, or copper ware is a first-order single empirical or mathematical connection proposition:

$$\text{bronze\_ware}(x) \leqslant^{-1} \text{bronze\_brass\_copper\_ware}(x) \tag{29.1}$$

A route in Fig. 29.1 from a leaf to the root is a first-order single logical connection proposition. For example, the root from tripod to bronze, brass, or copper ware is a first-order

single logical connection proposition:

{tripod(x)$\leqslant^{-1}$bronze_ware_for_rite(x)}$\wedge${bronze_ware_for_rite(x)$\leqslant^{-1}$bronze_ware(x)}$\wedge${bronze_ware(x)$\leqslant^{-1}$bronze_brass_copper_ware(x)}$\leqslant^{-1}${tripod(x)$\leqslant^{-1}$ bronze_brass_copper_ware(x)} (29.2)

Formula (29.2) is obtained as follows: the antecedent is the conjunction of the first-order single empirical or mathematical connection propositions denoted by the edges of the route, the consequent is an empirical or mathematical connection proposition, whose antecedent is the leaf of the route, whose consequent is the root.

A bottom-up second-level single quasi-expert system is used for theorem proving. For example, taking (29.2) as the major premise, taking the conjunction of (29.3), (29.4) and (29.1) as the minor premise, using second-level hypothetical inference, we can infer the conclusion (29.5).

tripod(x)$\leqslant^{-1}$bronze_ware_for_rite(x) (29.3)

bronze_ware_for_rite(x)$\leqslant^{-1}$bronze_ware(x) (29.4)

tripod(x)$\leqslant^{-1}$ bronze_brass_copper_ware(x) (29.5)

Bottom-up second-level single quasi-expert systems can be implemented as second-level single quasi-Prolog program.

29.4 Top-down second-level single quasi-expert systems

Each edge of Fig. 29.1 top-down is also a first-order single empirical or mathematical connection proposition. For example, the edge from bronze, brass, or copper ware to bronze ware is a first-order single empirical or mathematical connection proposition:

{bronze_brass_copper_ware(x)(pale(x))bronze_ware(x)} (29.6)

A route in Fig. 29.1 from the root to a leaf is a first-order single logical connection proposition. For example, the route from bronze, brass, or copper ware to tripod is a first-order single logical connection proposition:

{bronze_brass_copper_ware(x)(pale(x))bronze_ware(x)}$\wedge${bronze_ware(x)(for_rite(x))bronze_ware_for_rite(x)}$\wedge${bronze_ware_for_rite(x)(has_legs(x)$\wedge$as_container(x))tripod(x)}$\leqslant^{-1}${bronze_brass_copper_ware(x)(is)tripod(x)} (29.7)

Formula (29.7) is obtained as follows: the antecedent is the conjunction of the first-order single empirical or mathematical connection propositions denoted by the edges of the route, the consequent is a first-order single empirical or mathematical connection proposition, whose antecedent is the root, whose consequent is the leaf of the route.

Suppose a collector wants to know whether his collection is a tripod or not. Then he suppose

$$\neg\{\text{bronze\_brass\_copper\_ware}(x)(\text{is})\text{tripod}(x)\} \tag{29.8}$$

to be true. Formulas (29.7) and (29.8) make the second-level negative expression of hypothetical inference, inferring

$$\neg\{\text{pale}(x)\wedge\text{for\_rite}(x)\wedge\text{has\_legs}(x)\wedge\text{as\_container}(x)\} \tag{29.9}$$

Formula (29.9) is equivalent to

$$\neg\text{pale}(x)\vee/^{-1}\neg\text{for\_rite}(x)\vee/^{-1}\neg\text{has\_legs}(x)\vee/^{-1}\neg\text{as\_container}(x) \tag{29.10}$$

The system asks "is it pale?" The collector answers "Yes." The answer and (29.10) make disjunctive inference, inferring

$$\neg\text{for\_rite}(x)\vee/^{-1}\neg\text{has\_legs}(x)\vee/^{-1}\neg\text{as\_container}(x) \tag{29.11}$$

The system asks "is it for rite?" The collector answers "Yes." The answer and (29.11) make disjunctive inference, inferring

$$\neg\text{has\_legs}(x)\vee/^{-1}\neg\text{as\_container}(x) \tag{29.12}$$

The system asks "has it legs?" The collector answers "Yes." The answer and (29.12) make disjunctive inference, inferring

$$\neg\text{as\_container}(x) \tag{29.13}$$

The system asks "is it a container?" The collector answers "Yes." The answer and (29.13) make disjunctive inference, inferring contradiction. Thus, the supposition (29.8) does not hold, and the collection is a tripod.

A top-down second-level single quasi-expert system is used for classification. It can be implemented as second-level single quasi-Prolog program and goal. The first-order logical connection propositions such as (29.7) are the second-level conditional clauses, the first-order empirical or mathematical connection propositions such as (29.6) are the second-level unconditional clauses, and (29.8) is the second-level goal clause.

29.5 Improved version of top-down second-level single quasi-expert systems

Both the first-level single quasi-expert system of Section 29.2 and the top-down second-level single quasi-expert system of Section 29.4 can do bronze, brass, or copper ware classification, but their computational complexity are exponential with the height n of the tree given in Fig. 29.1. The improved version of the top-down second-level single quasi-expert system is shown in Fig. 29.2. It uses the left and right neighbors representation of a tree to denote the semantic network of Fig. 29.1. Its computational complexity is linear with the height n of the tree given in Fig. 29.1. The classification algorithm exerted on Fig. 29.2 is given in Fig. 29.3. The algorithm can exerts on any decision tree obtained from decision tree learning.

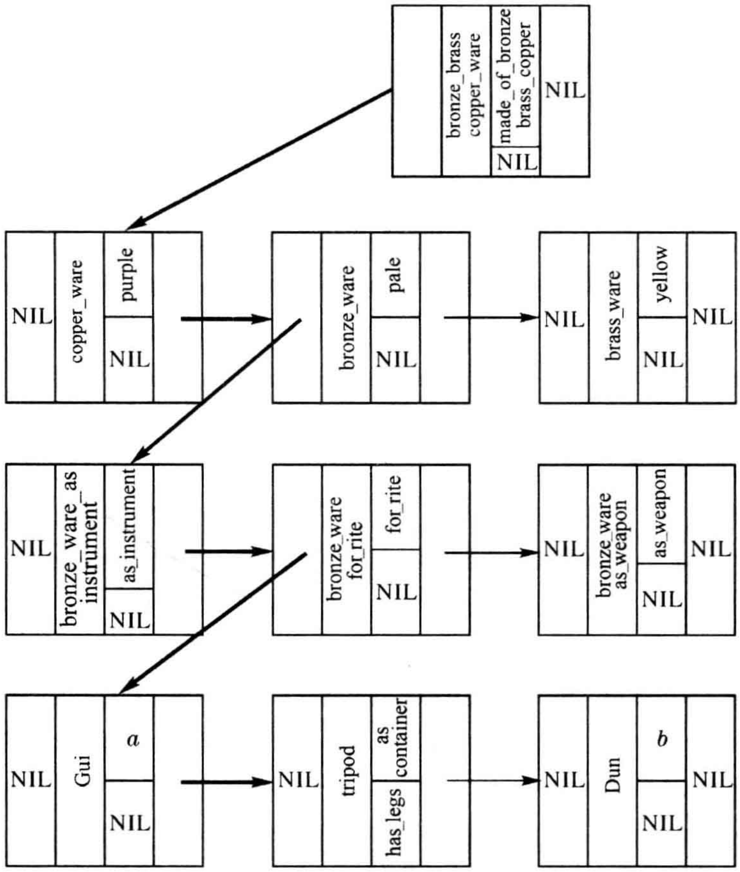


Fig. 29.2 Left and right neighbors representation of a tree

Suppose a collector wants to know whether his collection is a tripod or not. Then he exerts the algorithm shown in Fig. 29.3 to the data structure shown in Fig. 29.2. The route the algorithm goes through on the data structure is described by the thick arrows.

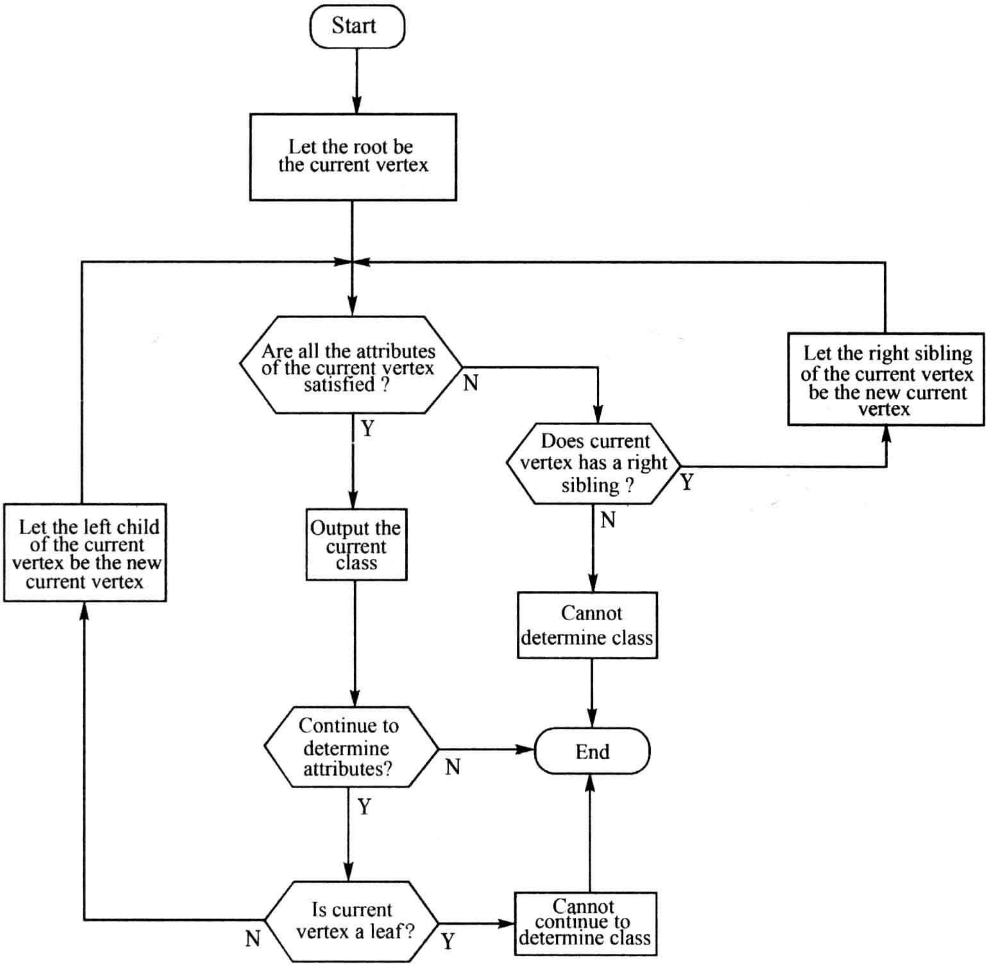


Fig. 29.3 Classification algorithm

Chapter 30 Transformation of second-level inference rule systems into second-level automated decomposition systems

30.1 Mutually-inversistic program verification

30.1.1 Introduction

In 1967, Floyd proposed invariant assertion method (Floyd, 1967), created the field of program verification. In 1969, Hoare proposed Hoare rules (Hoare, 1969), established the foundation for formal verification. In reading this section, readers should be familiar with invariant assertion method and Hoare rules. Mutually-inversistic program verification system is obtained by transforming the inference rules of Hoare rules into logical axioms, taking the second-level affirmative expression of hypothetical inference (abbreviated as second-level MP) as the sole inference rule. It is a second-level forward automated decomposition system, capable of proving the partial correctness of structured programs having sequence, selection, and loop structures. After assertions and verification conditions are established, the whole proof process is automated.

30.1.2 Hoare rules

Hoare rules include the following axiom and inference rules:

Axiom of assignment:

$\{P[\mathrm{a}/X]\}X{:=}\mathrm{a}\{P\}$

Rule of composition:

$\{P\}\mathrm{S}_0\{R\},\ \{R\}\mathrm{S}_1\{Q\}$

$\{P\}\mathrm{S}_0;\ \mathrm{S}_1\{Q\}$

Rule of condition:

$\{P\wedge R\}\mathrm{S}_0\{Q\},\ \{P\wedge\neg R\}\mathrm{S}_1\{Q\}$

$\{P\}\text{if } R \text{ then } \mathrm{S}_0 \text{ else } \mathrm{S}_1\{Q\}$

Rule of iteration:

$\{P\wedge R\}\mathrm{S}\{P\}$

$\{P\}$while R do S$\{P\wedge\neg R\}$

Rules of consequence:

$P\rightarrow R$, $\{R\}$S$\{Q\}$

$\{P\}$S$\{Q\}$

or

$\{P\}$S$\{R\}$, $R\rightarrow Q$

$\{P\}$S$\{Q\}$

30.1.3 Mutually-inversistic program verification

Mutually-inversistic program verification system is obtained by transforming the inference rules in Hoare rules into logical axioms, taking second-level MP as the sole inference rule. It has the following axioms:

Axiom of assignment:

$\{P[a/X]\}X$:=a$\{P\}$

Axiom of composition:

$\{P\}S_0\{R\}\wedge\{R\}S_1\{Q\}\leqslant^{-1}\{P\}S_0;\ S_1\{Q\}$

Axiom of condition:

$\{P\wedge R\}S_0\{Q\}\wedge\{P\wedge\neg R\}S_1\{Q\}\leqslant^{-1}\{P\}$if R then S_0 else $S_1\{Q\}$

Axiom of iteration:

$\{P\wedge R\}S\{P\}\leqslant^{-1}\{P\}$while R do S$\{P\wedge\neg R\}$

Axiom of left consequence:

$(P\leqslant^{-1}R)\wedge\{R\}S\{Q\}\leqslant^{-1}\{P\}S\{Q\}$

Axiom of right consequence:

$\{P\}S\{R\}\wedge(R\leqslant^{-1}Q)\leqslant^{-1}\{P\}S\{Q\}$

S in the invariant statement $\{P\}$S$\{Q\}$ is a program segment, which is as small as a single statement, as large as the whole program. P is its preassertion, Q is its postassertion. When S is the whole program, the preassertion is called the input assertion, the postassertion is called the output assertion. $\{P\}$S$\{Q\}$ can be regarded as a single empirical or mathematical connection proposition $P\leqslant^{-1}Q$. Suppose P is true. If we can infer that Q is true, then we determine that $P\leqslant^{-1}Q$ is true. Likewise, suppose the preassertion P is true before the execution of the program segment S. If after the execution of S, S terminates, and the post-assertion Q is true, then we determine that the invariant statement $\{P\}$S$\{Q\}$ is true, or that the program segment S is partially correct. A single empirical or mathematical connection proposition can be proved by second-level hypothetical inference. In this section, the partial correctness of the invariant statement is proved by the second-level MP.

30.1.4 Constituents of the mutually-inversistic program verification system

The mutually-inversistic program verification system is composed of five parts: the program to be verified, the assertions, the verification conditions, the six axioms mentioned in Section 30.1.3, the proof algorithm. The assertions and verification conditions are established like the invariant assertion method proposed by Floyd, that is, make break points in the proper places of the program, establish assertions at the break points, view the program segment between two break points as a route, according to the pass condition, preassertion, and postassertion to establish the verification condition.

30.1.5 Proof algorithm

The proof algorithm is as follows: read each assignment statement in turn; attach the preassertion and postassertion to it to form an invariant statement; carry out consequence inference (which will be explained later); see whether two invariant statements can be combined by one of composition combination, while loop combination, or if-then-else combination; if they can, then do it; then carry out consequence inference; combinations are usually nested; if combination cannot be carried out, then read the next assignment statement, and repeat the above process; if no more assignment statement, then decide whether the invariant statement deduced is the same as the preassertion established, the program to be verified, the postassertion established; if so, then the program to be verified is partially correct, otherwise, it is not; and the algorithm terminates.

The consequence inference is carried out as follows: take the invariant statement as one minor premise, take a verification condition as the other minor premise, take the axiom of the left consequence or the axiom of the right consequence as the major premise, use second-level MP, infer a new invariant statement. The three combinations are easy to understand, we will discuss them in the example that follows.

The flow chart of the proof algorithm is shown in Fig. 30.1, its end frame is expanded in Fig. 30.2.

30.1.6 An example

Example 30.1: Prove the partial correctness of the program of seeking the greatest common divisor z of two nonnegative integers x and y.

Solution: The program of the problem is as follows:

START

$(x, y) \leftarrow (x_0, y_0)$;

while $x \neq 0$ do

if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)$;

$z \leftarrow y$

HALT

(1) Establish assertions. Suppose the initial values of x and y are x_0 and y_0, then the following input, output, and invariant assertions can be established:

$\varphi(x)$: $x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)$

$\psi(x, z)$: $z = \gcd(x_0, y_0)$

$p(x, y)$: $x \geqslant 0 \wedge y \geqslant 0 \wedge (x \neq 0 \vee y \neq 0) \wedge \gcd(x, y) = \gcd(x_0, y_0)$

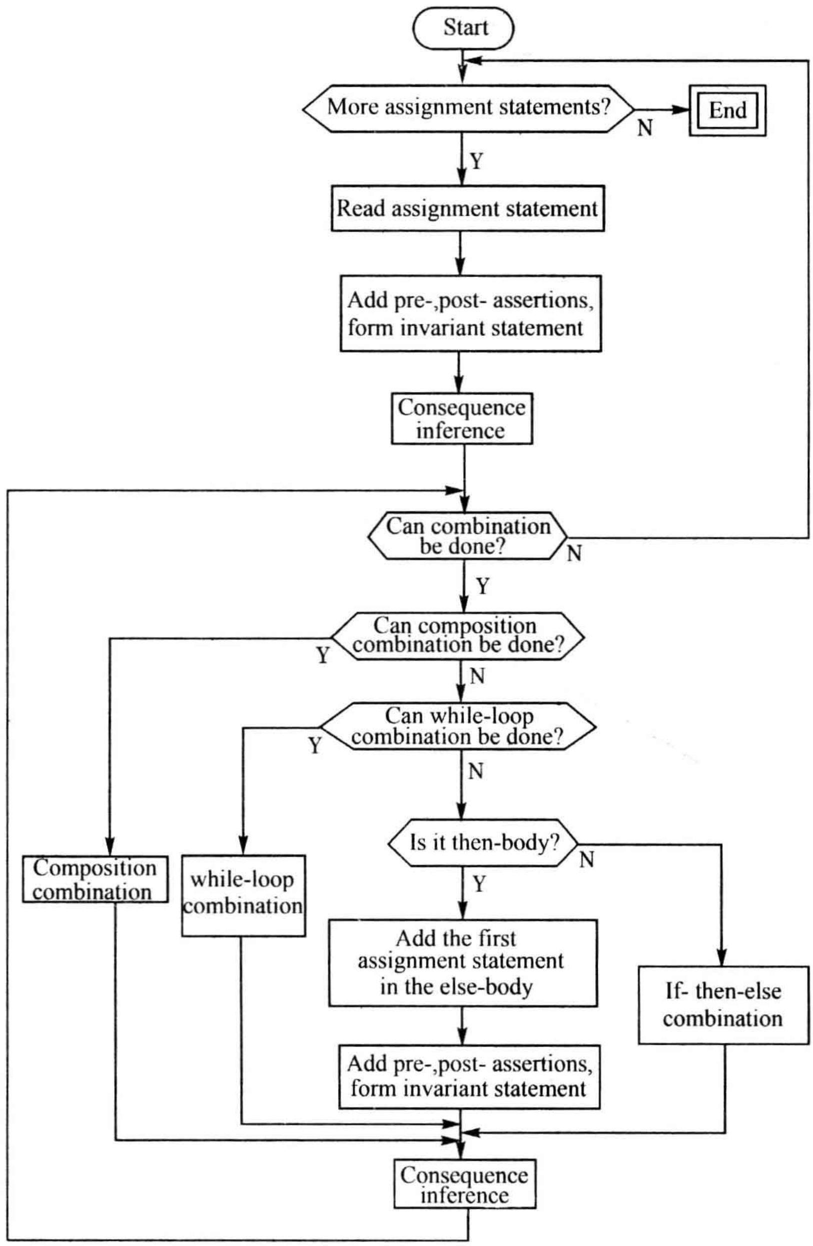


Fig. 30.1 Proof algorithm

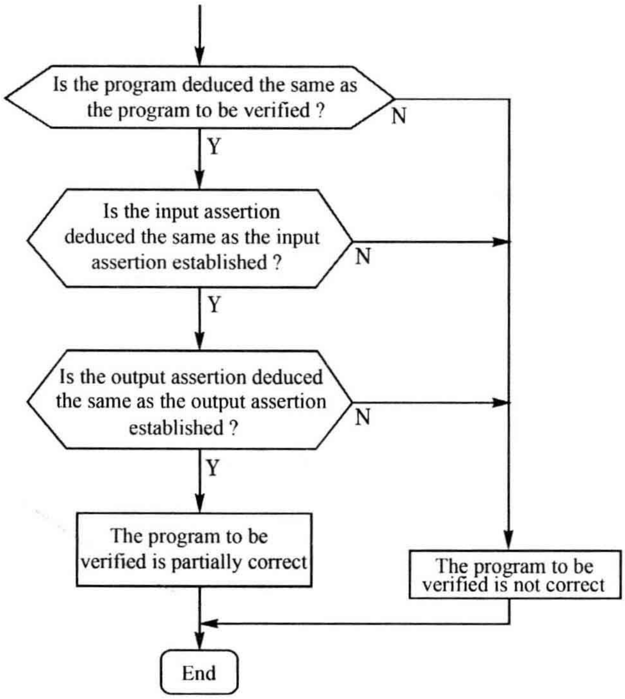


Fig. 30.2 Expansion of the end frame

(2) Attach the assertions to the program, obtaining

START

$\{x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)\}$

$(x, y) \leftarrow (x_0, y_0);$

while $x \neq 0$ do

L: $\{p(x, y)\}$

if $y > x$ then $y \leftarrow y - x$

else$(s, x, y) \leftarrow (x, y, s);$

$\{y = \gcd(x_0, y_0)\}$

$z \leftarrow y$

$\{z = \gcd(x_0, y_0)\}$

HALT

(3) Establish verification conditions. First, suppose the input assertion hold, and we want to prove that when the program execution first reaches L, the invariant assertion holds. As, at this time, the initial values of x and y are x_0 and y_0, we obtain Verification condition 1:

$$\{x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)\} \leqslant^{-1} p(x_0, y_0)$$

Secondly, we want to prove that when the invariant assertion holds and the control go back to L through the loop, the invariant assertion still holds. We know that the

condition for the loop to continue is $x \neq 0$, and two possibilities exist for the control to execute the conditional statement in the loop: one is when $y > x$ holds; i.e., the control go back to L along the then route. At this time, we can obtain

Verification condition 2:

$$\{p(x, y) \wedge x \neq 0 \wedge y > x\} \leqslant^{-1} p(x, y-x)$$

The other is when $y < x$ holds; i.e., the control go back to L along the else route. At this time, we can obtain

Verification condition 3:

$$\{p(x, y) \wedge x \neq 0 \wedge y < x\} \leqslant^{-1} p(y, x)$$

Lastly, we want to prove that if the invariant assertion holds, and the control go out of the loop, then the output assertion holds. As the condition for the control go out of the loop is $x=0$, we can obtain

Verification condition 4:

$$\{p(x, y) \wedge x=0\} \leqslant^{-1} y=\gcd(x_0, y_0)$$

(4) Prove the verification conditions. Omitted.

(5) Prove the partial correctness of the program.

(i) Read the first assignment statement and attach the preassertion and postassertion.

$$\{p(x_0, y_0)\}(x, y) \leftarrow (x_0, y_0)\{p(x, y)\}$$

(ii) Consequence inference. Take the axiom of the left consequence as the major premise, take the verification condition 1 and (i) as the minor premise, use second-level MP, infer

$$\{x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)\}(x, y) \leftarrow (x_0, y_0)\{p(x, y)\}$$

(iii) Read the second assignment statement and attach the preassertion and postassertion.

$$\{p(x, y-x)\} y \leftarrow y-x \{p(x, y)\}$$

(iv) Consequence inference. Take the axiom of the left consequence as the major premise, take the verification condition 2 and (iii) as the minor premise, use second-level MP, infer

$$\{p(x, y) \wedge x \neq 0 \wedge y > x\} y \leftarrow y-x \{p(x, y)\}$$

(v) (iv) is the then-body, read the first assignment statement in the else-body and attach the preassertion and postassertion.

$$\{p(y, x)\}(s, x, y) \leftarrow (x, y, s)\{p(x, y)\}$$

(vi) Consequence inference. Take the axiom of the left consequence as the major premise, take the verification condition 3 and (v) as the minor premise, use second-level MP, infer

$$\{p(x, y) \wedge x \neq 0 \wedge y < x\}(s, x, y) \leftarrow (x, y, s)\{p(x, y)\}$$

(vii) If-then-else combination. Take the axiom of condition as the major premise, take

(iv) and (vi) as the minor premise, use second-level MP, infer

$\{p(x, y) \wedge x \neq 0\}$ if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)\{p(x, y)\}$

(viii) While loop combination. Take the axiom of iteration as the major premise, take (vii) as the minor premise, use second-level MP, infer

$\{p(x, y)\}$ while $x \neq 0$ do

if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)\{p(x, y) \wedge x = 0\}$

(ix) Consequence inference. Take the axiom of right consequence as the major premise, take (viii) and the verification condition 4 as the minor premise, use second-level MP, infer

$\{p(x, y)\}$ while $x \neq 0$ do

if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)\{y = \gcd(x_0, y_0)\}$

(x) Composition combination. Take the axiom of composition as the major premise, take (ii) and (ix) as the minor premise, use second-level MP, infer

$\{x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)\}$

$(x, y) \leftarrow (x_0, y_0)$

while $x \neq 0$ do

if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)\{y = \gcd(x_0, y_0)\}$

(xi) Read the fourth assignment statement and attach the preassertion and postassertion.

$\{y = \gcd(x_0, y_0)\} z \leftarrow y \{z = \gcd(x_0, y_0)\}$

(xii) Composition combination. Take the axiom of composition as the major premise, take (x) and (xi) as the minor premise, use second-level MP, infer

$\{x_0 \geqslant 0 \wedge y_0 \geqslant \wedge (x_0 \neq 0 \vee y_0 \neq 0)\}$

$(x, y) \leftarrow (x_0, y_0)$

while $x \neq 0$ do

if $y > x$ then $y \leftarrow y - x$

else $(s, x, y) \leftarrow (x, y, s)$

$z \leftarrow y \{z = \gcd(x_0, y_0)\}$

(xiii) The program is partially correct. End.

30.2 Automated deduction system of functional dependency of relational database

The system is obtained by transforming the inference rules of Armstrong axiomatic

system of relational database into the logical axioms; e.g., the transitive law is transformed into $\{X\to Y\}\wedge\{Y\to Z\}\leqslant^{-1}\{X\to Z\}$, and taking second-level MP and the rule of conjunction as the sole inference rules. For example, the process of deducing SNO→SLOC from SNO→SDEPT and SDEPT→SLOC is as follows:

(1) SNO→SDEPT	P
(2) SDEPT→SLOC	P
(3) {SNO→SDEPT}∧{SDEPT→SLOC}	T(1)(2)Rule of conjunction
(4) $\{X\to Y\}\wedge\{Y\to Z\}\leqslant^{-1}\{X\to Z\}$	P
(5) SNO→SLOC	T(3)(4)Second-level MP

30.3 Mutually-inversistic operational semantics

Mutually-inversistic operational semantics is obtained by transforming the semantic rules in the traditional operational semantics into logical axioms, taking second-level MP and the rule of conjunction as the sole inference rules. It aims at developing the formal semantics into the automated semantics. The semantic rule for addition:

$$\frac{\langle a_0, \sigma\rangle\to n_0,\ \langle a_1, \sigma\rangle\to n_1}{\langle a_0+a_1, \sigma\rangle\to n_2}\quad \text{the addition of } n_0 \text{ and } n_1 \text{ equals } n_2$$

can be transformed into the logical axiom:

$$\{\langle a_0, \sigma\rangle\leqslant^{-1}n_0\wedge \langle a_1, \sigma\rangle\leqslant^{-1}n_1\}\leqslant^{-1}\langle a_0+a_1, \sigma\rangle\leqslant^{-1}n_2.$$

After the logical axiom and the two single empirical or mathematical theorems:

$$\langle 5, \sigma_0\rangle\leqslant^{-1}5$$

and

$$\langle 7, \sigma_0\rangle\leqslant^{-1}7$$

are known, the process of inferring $\langle 5+7, \sigma_0\rangle\leqslant^{-1}12$ is as follows:

(1) $\langle 5, \sigma_0\rangle\leqslant^{-1}5$	P
(2) $\langle 7, \sigma_0\rangle\leqslant^{-1}7$	P
(3) $\langle 5, \sigma_0\rangle\leqslant^{-1}5\wedge\langle 7, \sigma_0\rangle\leqslant^{-1}7$	T(1)(2)Rule of conjunction
(4) $\{\langle a_0, \sigma\rangle\leqslant^{-1}n_0\wedge \langle a_1, \sigma\rangle\leqslant^{-1}n_1\}\leqslant^{-1}\langle a_0+a_1, \sigma\rangle\leqslant^{-1}n_2$	P
(5) $\langle 5+7, \sigma_0\rangle\leqslant^{-1}12$	T(3)(4)second-level MP

30.4 Second-level hypothetical inference based LK system of proof theory

LK system of proof theory is a second-level inference rule system. Some of the inference rules are as follows:

$\Gamma \to \Delta, F(\mathrm{t})$

∃: right: ----------------

$\Gamma \to \Delta, \exists_x F(x)$

$\Gamma \to \Delta, D$

¬: left: --------------

$\neg D, \Gamma \to \Delta$

$\Gamma, C, D, \Pi \to \Delta$

exchange: left: ------------------

$\Gamma, D, C, \Pi \to \Delta$

$D, \Gamma \to \Delta$

¬right: --------------

$\Gamma \to \Delta, \neg D$

$\Gamma \to \Delta, F(\mathrm{a})$

∀: right: ----------------

$\Gamma \to \Delta, \forall_x F(x)$

The formal proof of the logical theorem $\neg\exists_x F(x) \to \forall_y \neg F(y)$ is as follows:

$F(\mathrm{a}) \to F(\mathrm{a})$

∃: right ----------------

$F(\mathrm{a}) \to \exists_x F(x)$

¬: left ----------------

$\neg\exists_x F(x), F(\mathrm{a}) \to$

exchange: left ---------------------

$F(\mathrm{a}), \neg\exists_x F(x) \to$

¬:right -----------------------

$\neg\exists_x F(x) \to \neg F(\mathrm{a})$

∀: right --------------------------

$\neg\exists_x F(x) \to \forall_y \neg F(y)$

Transforming the inference rules into logical axioms, taking second-level hypothetical inference as the inference rule, we can automate the theorem proving of this system. The second-level single quasi-Prolog program and goal for proving $\neg\exists_x F(x) \to \forall_y \neg F(y)$ is as follows:

(1) $F(\mathrm{a}) \leftarrow F(\mathrm{a})$.

(2) $\exists_x F(x) \leftarrow \Gamma$===>>>$F(\mathrm{t}) \leftarrow \Gamma$. (∃: right axiom)

(3) $\leftarrow \neg D, \Gamma$===>>>$D \leftarrow \Gamma$. (¬: left axiom)

(4) $\leftarrow D, C$===>>>$\leftarrow C, D$. (exchange: left axiom)

(5) $\neg D \leftarrow \Gamma$===>>>$\leftarrow D, \Gamma$. (¬: right axiom)

(6) $\forall_x \neg F(x) \leftarrow \Gamma$===>>>$\neg F(\mathrm{a}) \leftarrow \Gamma$. (∀: right axiom)

(7) $?\text{-}\forall_y \neg F(y) \leftarrow \neg \exists_x F(x)$.

In the program, (2) through (6) are the logical axioms transformed from the inference rules. The second-level SLD tree of the program and goal is shown in Fig. 30.3.

$?\text{-}\forall y \neg F(y) \leftarrow \neg\exists x F(x).$

(6) $\{\Gamma_1/\neg\exists x F(x)\}$

$?\text{-}\neg F(\mathrm{a}) \leftarrow \neg\exists x F(x).$

(5) $\{D_2/F(a), \Gamma_2/\neg\exists x F(x)\}$

$?\text{-} \leftarrow F(\mathrm{a}), \neg\exists x F(x).$

(4) $\{D_3/F(a), C_3/\neg\exists x F(x)\}$

$?\text{-} \leftarrow \neg\exists x F(x), F(a).$

(3) $\{D_4/\exists x F(x), \Gamma_4/F(a)\}$

$?\text{-}\exists x F(x) \leftarrow F(a).$

(2) $\{\Gamma_5/F(a)\}$

$?\text{-}F(t) \leftarrow F(a).$

(1) $\{t/a\}$

$\square$

Fig. 30.3 Second-level SLD tree for LK system

30.5 Second-level hypothetical inference based propositional logic of natural deduction system

The propositional logic of natural deduction system has such inference rules as distributive law $A\wedge(B\vee C)\leftrightarrow A\wedge B\vee A\wedge C$ and dilemma $(A\rightarrow C)\wedge(B\rightarrow D)\wedge(A\vee B)\rightarrow(C\vee D)$. The proof from the premises

$P\wedge Q\rightarrow R\wedge S$

$P\wedge(Q\vee R)$

$P\wedge R\rightarrow R\wedge T$

to the conclusion

$R\wedge(S\vee T)$

is as follows:

(1) $P \wedge Q \to R \wedge S$ P

(2) $P \wedge (Q \vee R)$ P

(3) $P \wedge R \to R \wedge T$ P

(4) $(P \wedge Q) \vee (P \wedge R)$ T(2)Distributive law

(5) $(R \wedge S) \vee (R \wedge T)$ T(1)(3)(4)Dilemma

(6) $R \wedge (S \vee T)$ T(5)Distributive law

Transforming the inference rules into logical axioms, taking second-level hypothetical inference as the inference rule, we can automate the proof. After the transformation, the second-level single quasi-Prolog program and goal are as follows:

(1) $R \wedge S \leftarrow P \wedge Q$.

(2) $P \wedge (Q \vee R)$.

(3) $R \wedge T \leftarrow P \wedge R$.

(4) $A \wedge (B \vee C) ===>>> A \wedge B \vee A \wedge C$.

(5) $A \wedge B \vee A \wedge C ===>>> A \wedge (B \vee C)$.

(6) $C \vee D ===>>> C \leftarrow A,\ D \leftarrow B,\ A \vee B$.

(7) ?-$R \wedge (S \vee T)$.

In the program, (4) through (6) are the logical axioms transformed from the distributive law and dilemma. The second-level SLD tree of the program and goal is shown in Fig. 30.4.

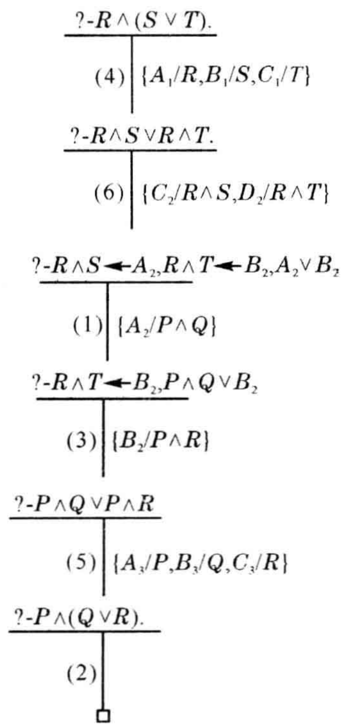


Fig. 30.4 Second-level SLD tree for natural deduction system

30.6 Second-level hypothetical inference based nonassociative Lambek calculus of categorial grammar

Categorial grammar deems that every word can be included into certain syntactic type according to its functionality in a sentence. For example, the syntactic type of John is N(noun), denoted by John:N; the syntactic type of works is N\S, denoted by works: N\S, where S is a sentence. In the system, there is \E rule:

$\Gamma \Rightarrow \varphi: A \quad \Delta \Rightarrow \chi: A\backslash B$

-------------------------\E

$\Gamma, \Delta \Rightarrow \chi\varphi: B$

It says: if the syntactic type of phrase χ is $A\backslash B$, the syntactic type of the previous word φ is A, then the syntactic type of $\chi\varphi$ is B. Applying the rule to John:N and works: N\S, we obtain John works:S; i.e., *John works* is a sentence. In the system, there is also /E rule:

$\Delta \Rightarrow \chi: B/A \quad \Gamma \Rightarrow \varphi: A$

-------------------------/E

$\Delta, \Gamma \Rightarrow \chi\varphi: B$

It says: if the syntactic type of phrase χ is B/A, the syntactic type of the word φ after it is A, then the syntactic type of $\chi\varphi$ is B. Now, we use /E to analyze that *every man walks* is a sentence. Suppose we have

every: ((S)/(N\S))/CN

man: CN

walks: (N\S)

where CN means common noun. The deduction that *every man walks* is a sentence is as follows:

```
every              man            walks
-----------------  ------
((S/(N\S))/CN      CN
--------------------------------/E  -------
         (S)/(N\S)                  (N\S)
----------------------------------------------------/E
                   (S)
```

Transforming the inference rules into logical axioms, taking second-level hypothetical inference as the inference rule, we can automate the deduction. The second-level single quasi-Prolog program and goal of the above deduction is as follows:

(1) every: ((S)/(N\S))/CN.

(2) man: CN.

(3) walks: (N\S).

(4) right: $\chi\varphi$: B===>>>χ: $(B)/A$, φ: A.

(5) ?-every man walks: S.

In the program, (4) is /E axiom obtained from /E rule, where right means that φ in the head unifies first. The second-level SLD tree of the program and goal is shown in Fig. 30.5.

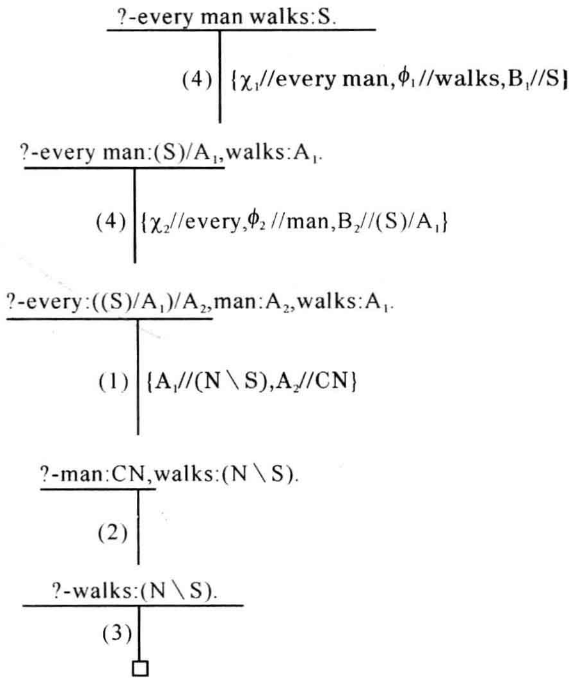


Fig. 30.5 Second-level SLD tree for *every man walks*

Chapter 31 Applications of First-Level Hypothetical Inference

31.1 Mutually-inversistic natural language understanding

31.1.1 First-level bottom-up parser

31.1.1.1 Introduction to first-level bottom-up parser

The rule of a deductive bottom-up parser should have more heads, one body. The parser should do shift and reduce operations, reducing the heads to the body. The rule of Prolog has one head, more bodies. Prolog is suitable for deductive top-down parsing and left-corner parsing, but is unsuitable for deductive bottom-up parsing. Yet, the techniques adopted by the so-called deductive bottom-up parser such as agent-driven, chart-based parsing, memoing, magic sets are all implemented by Prolog. Such so-called deductive bottom-up parsers as bottom-up Earley deduction, selective magic HPSG parser, XSB, DyALog are all implemented by Prolog. So, they are not genuine deductive bottom-up parsers.

First-level bottom-up parser proposed in this section is different from Prolog. Its rules have more heads, one body. It can do shift and reduce operations, reducing the heads to the body. So, it is a genuine deductive bottom-up parser.

The grammar rule of a language is in the form of *NP, VP*$\leftarrow$*S*, which says that *NP* and *VP* can be inferred from *S*. The rule clause of first-level bottom-up parser is in the form of *NP, VP: -S*, which also says that *NP* and *VP* can be inferred from *S*, just the same. The Prolog rule clause is in the form of sentence(s_0, s): -noun_phrase(s_0, s_1), verb_phrase(s_1, s), which says that sentence(s_0, s) can be inferred from noun_phrase(s_0, s_1) and verb_phrase(s_1, s), just a reverse in direction.

First-level bottom-up parser takes the grammar rules as theorems, take the first-level negative expression of hypothetical inference as the inference rule. In this way, the parser can be automated.

The construction of first-level bottom-up parser is straightforward: the grammar rules are the conditional clauses, the sentence to be parsed is the goal clause, the start symbol is the unconditional clause. First-level bottom-up parser is suitable for semantic analysis.

31.1.1.2 First-level bottom-up parser

Suppose we have the following grammar rules:

NP VP←S.

ART ADJ N←NP.

ART N←NP.

ADJ N←NP.

AUX VP←VP.

V NP←VP.

the←ART.

large←ADJ.

can←AUX.

can←N.

can←V.

hold←V.

hold←N.

water←N.

water←V.

And we want to parse the sentence: *the large can can hold the water.* The first-level bottom-up parser program and initial goal is as follows:

(1) S.

(2) NP VP:-S.

(3) ART ADJ N:-NP.

(4) ART N:-NP.

(5) ADJ N:-NP.

(6) AUX VP:-VP.

(7) V NP:-VP.

(8) the:-ART.

(9) large:-ADJ.

(10) can:-AUX.

(11) can:-N.

(12) can:-V.

(13) hold:-V.

(14) hold:-N.

(15) water:-N.

(16) water:-V.

(17) ?-the large can can hold the water.

Clause (1) is the start symbol and the unit clause. Clauses (2) through (16) are rule clauses. Clause (17) is the initial goal (we call it initial goal, because in the SLD-like tree the goal changes constantly). Clauses (1) through (7) are the non-terminal program. Clauses (8)

through (16) are the terminal program. The rules of non-terminal program have more heads, one body. The syntactic variables don't take arguments. First-level bottom-up parser adopts linear resolution, i.e. one resolver is the goal clause, the other is the program clause. The search strategy is top-down, depth-first plus backtracking, not from left to right but the handle. This means that the unification is not with the leftmost subgoal but is with the handle.

?-the large can can hold the water.

(8)

?-ART large can can hold the water.

(9)

?-ART ADJ can can hold the water.

(11)

?-ART ADJ N can hold the water.

(3)

?-NP can hold the water.

(10)

?-NP AUX hold the water.

(13)

?-NP AUX V the water.

(8)

?-NP AUX V ART water.

(15)

?-NP AUX V ART N.

(4)

?-NP AUX V NP.

(7)

?-NP AUX VP.

(6)

?-NP VP.

(2)

?- S.

(1)

□

Fig. 31.1 SLD-like tree for first-level bottom-up parser

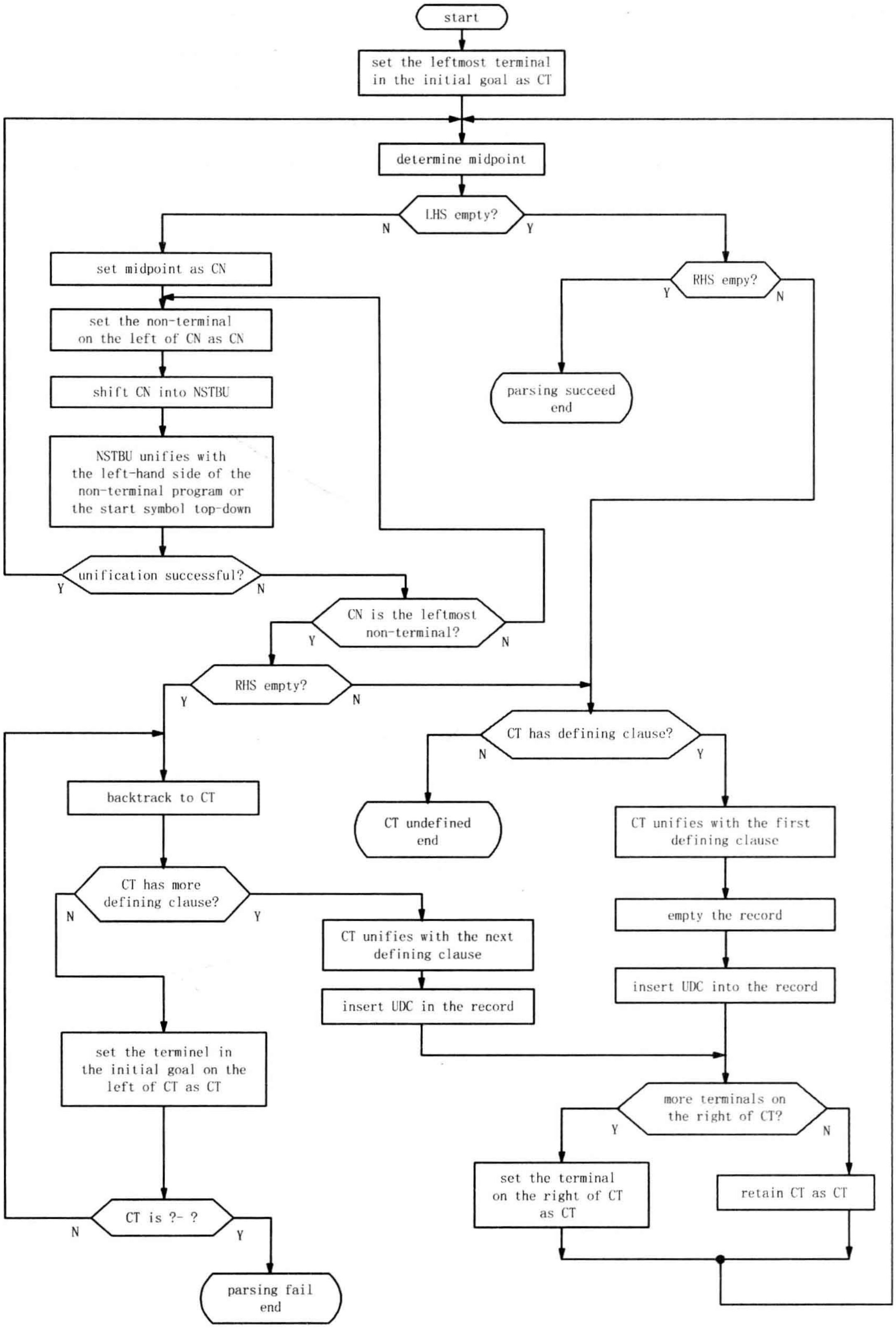


Fig. 31.2 Flowchart of first-level bottom-up parser

First-level bottom-up parser can parse context-free grammar. The SLD-like tree of the program and the initial goal is shown in Fig. 31.1. In Fig. 31.1, all backtracking are omitted. A handle is either the non-terminal string to be unified (NSTBU) or the current terminal (CT). The handles are underlined. From Fig. 31.1 we see that at any stage of the parsing, the goal has a midpoint. The left-hand side (LHS) of the midpoint is all the non-terminals, the right-hand side (RHS) of the midpoint is all the terminals. At the beginning of the parsing, midpoint is at the left end. Then, the leftmost terminal to the right of the midpoint is taken as the CT, and is unified with the terminal program. Then, adjust the midpoint. When LHS is not empty, take the non-terminal on the left of NSTBU as the current non-terminal (CN), shift CN to NSTBU and do unification; keep doing this until no more can be done. Then, if RHS is not empty, unifies CT. If RHS is empty, then backtracks. If both LHS and RHS are empty, then terminates with success. If backtracking to ?-, then terminates with failure. Each terminal keeps a record of unified defining clause (UDC) to keep track of the clauses unified. In Fig. 31.1, all backtracking are omitted, and the midpoint moves steadily from the left end to the right end. In the circumstances where backtracking is considered, the midpoint moves back and forth between the two ends. The flowchart of first-level bottom-up parser is shown in Fig. 31.2.

31.1.2 First-level bottom-up parser based natural language understanding

Suppose we have the following grammar rules:

NP, VP←S

John←NP

reads←VP.

and we want to deduce *John reads*. The first-level bottom-up parser program and goal are as follows:

(1) *S.*

(2) *NP, VP: -S.*

(3) *John: -NP.*

(4) *reads: -VP.*

(5) *?-John reads.*

The syntactic SLD-like tree of the program and goal is shown in Fig. 31.3.

In order to understand the sentence, we need the following lexical ontology:

Name: is(x_1, name)$\wedge$grammar_category(x_1, NP)$\wedge$collocations(x_1, {reads, runs, laughs})$\wedge${case(x_1, agent)$\oplus$case(x_1, theme)}$\wedge$hypernym(x_1, person) (31.1)

Reads: is(y_0, reads)$\wedge$grammar_category(y_0, VP)$\wedge$agent(y_0, y_1)$\wedge$case(y_1, agent)$\wedge$ hypernym(y_1, person) (31.2)

?-John reads.

(3)

?-NP, reads.

(4)

?-NP, VP.

(2)

?-S.

(1)

□

Fig. 31.3 Syntactic SLD-like tree

The predicates in the lexical ontology are all binary predicates, easy to be described by the ontology description language. Mutually-inversistic natural language understanding adopts five semantic analysis techniques: lexical semantics driven, syntax-directed, thematic role, selectional restrictions, and collocations. Agent role and theme role are thematic role. Hypernym predicates are selectional restrictions. Collocation predicates are collocations.

The system is syntax-directed. The first semantic step corresponding to the first syntactic step in Fig. 31.3 is that not only John is reduced to NP but also (31.1) is read and name is substituted by John, as is shown in the first step of the semantic SLD-like tree of Fig. 31.4.

?-John reads.

(3)

?- is(x_1, John)$\wedge$grammar_category(x_1, NP)$\wedge$collocations(x_1, {reads, runs, laughs})$\wedge$ {case(x_1, agent)$\oplus$case(x_1, theme)}$\wedge$hypernym(x_1, person), reads.

(4)

?- is(x_1, John)$\wedge$grammar_category(x_1, NP)$\wedge$collocations(x_1, {reads, runs, laughs})$\wedge$ {case(x_1, agent)$\oplus$case(x_1, theme)}$\wedge$hypernym(x_1, person), is(y_0, reads)$\wedge$grammar_category(y_0, VP)$\wedge$agent(y_0, y_1)$\wedge$case(y_1, agent)$\wedge$hypernym(y_1, person).

(2) $\{y_1/x_1\}$

?-is(x_1y_0, John reads)$\wedge$grammar_category(x_1y_0, S)$\wedge$agent(y_0, x_1)$\wedge$case (x_1, agent)$\wedge$ hypernym(x_1, person).

(1)

□

Fig. 31.4 Semantic SLD-like tree

The second semantic step corresponding to the second syntactic step in Fig. 31.3 is that not only *reads* is reduced to VP but also (31.2) is read, as is shown in the second step of Fig. 31.4. The third semantic step corresponding to the third syntactic step in Fig. 31.3 is that not only NP and VP are reduced to S but also (31.1) and (31.2) are unified when y_l is substituted by x_l, as is shown in the third step of Fig. 31.4.

In the third semantic step in Fig. 31.4, in testing hypernym(x_l, person), the system consults WordNet.

31.2 Mutually-inversistic hardware verifier

31.2.1 Introduction to mutually-inversistic hardware verifier

One method for hardware verification is the theorem prover. We hope that the theorem provers are automated without human intervention, because human beings may make mistakes. The major theorem provers are ACL2, PVS, HOL, etc., they are all interactive theorem provers, they all need human intervention. The reasons for this are twofold. First, they are all general-purpose theorem provers, used in hardware verification, software verification, and mathematical theorem proving. Secondly, the majority of them are based on classical logic. General-purposeness requires that they have mant proof methods: induction and deduction. Induction is divided into two steps: basis and induction, they all need human intervention. There are two kind of deduction. The first kind is that suppose A is true, if we can infer B is bound to be true, then we infer A→B. In HOL, this kind of deduction is called disch. Disch is divided into two steps: first, infer B from A, secondly, prove A→B to be true. They all need human intervention. The second kind is from A→B being true and A being true to infer B being true. In HOL, this kind of deduction is called the rule of detachment. General-purposeness requires that they have many inference rules. It is easy for a human being to choose when to use which rule. But it is hard for a computer to choose when to use which rule. General-purposeness requires that they have various axioms, not suitable to deal with uniformly. For example, in HOL, there is both such axiom as |-!*t*.(*t*=T)∨(*t*=F) and such axiom as |-!*t1t2*.(*t1*==>*t2*)==>(*t2*==>*t1*)==>(*t1*=*t2*). PVS and HOL are based on classical logic which has quantifiers, and it needs human intervention to eliminate quantifiers. Because of the above reasons, ACL2, PVS, HOL, etc., are all interactive theorem provers.

Mutually-inversistic hardware verifier is a special-purpose hardware verifier, it is based on mutually-inversistic logic. Special-purposeness means that it has only one proof method: the first-level affirmative expression of hypothetical inference. Every time the system proves a theorem, the system uses it. Special-purposeness means that it has only one inference

rule: the first-level affirmative expression of hypothetical inference. Every time the system makes inference, the system uses it. Special-purposeness means that its axioms are as uniform as the rule clauses of Prolog. Mutually-inversistic logic is quantifier-free. Because of these reasons, mutually-inversistic hardware verifier is an automated theorem prover.

31.2.2 Automated verification of combinational logic circuits

We take the automated verification of an exclusive-or gate as an example to show that of combinational logic circuit and to show the working principle of mutually-inversistic hardware verifier.

The structural description of an exclusive-or gate is shown in Fig. 31.5.

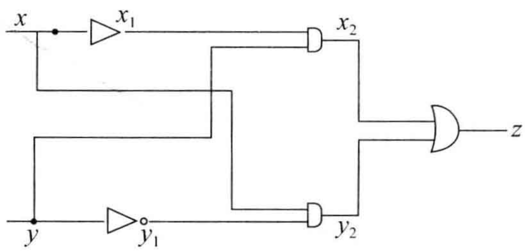


Fig. 31.5 Structural description of an exclusive-or gate

We want to verify whether the structural description implements the behavioral description: $(\neg x \wedge y) \vee (x \wedge \neg y)$. The verification is carried out by transitions from the structural description to the behavioral description. According to Fig. 31.5, the structural formula of the exclusive-or gate is:

$$\text{NOT}(x, x_1), \text{NOT}(y, y_1), \text{AND}(x_1, y, x_2), \text{AND}(x, y_1, y_2), \text{OR}(x_2, y_2, z) \quad (31.3)$$

where the commas denote $\wedge$. The transition formulas from the structural description to the behavioral description are:

$$\text{NOT}(in, out) \text{-:}\ out = \neg in^* \quad (31.4)$$

$$\text{AND}(in_1, in_2, out) \text{-:}\ out = in_1 \wedge in_2^* \quad (31.5)$$

$$\text{OR}(in_1, in_2, out) \text{-:}\ out = in_1 \vee in_2^* \quad (31.6)$$

where -: denotes $\leqslant^{-1}$, each antecedent is a structural subformula, and each consequent marked with $*$ is a behavioral subformula. The behavioral subformula to be verified is:

$$z = (\neg x \wedge y) \vee (x \wedge \neg y)^* \quad (31.7)$$

The verification starts from the structural formula (31.3), in which the structural subformulas from left to right respectively unify with the antecedents of the transition formulas top-down respectively. If the unification is successful, then the consequent of the corresponding transition formula is inferred (it is a behavioral subformula). The structural subformula involved is a zeroth-order fact proposition, the transition formula involved is an

empirical or mathematical axiom, the inference made is the first-level affirmative expression of hypothetical inference. If the behavioral subformula to be verified is inferred at last, then the verification is successful, otherwise, it is a failure. The first-level forward inference tree is shown in Fig. 31.6.

$$\frac{NOT(x,x_1),}{(31.4)\mid\{in/x,out/x_1\}}\qquad NOT(y,y_1), AND(x_1,y.x_2), AND(x,y_1,y_2).OR(x_2,y_2.z).$$

$$x_1=\neg x^*, y_1 \quad \frac{NOT(y,y_1),}{(31.4)\mid\{in/y,out/y_1\}}\qquad AND(\neg x,y.x_2), AND(x,y_1,y_2).OR(x_2,y_2,z).$$

$$x_1=\neg x^*, y_1=\neg y^*, \quad \frac{AND(\neg x,y,x_2),}{(31.5)\mid\{in_1/\neg x,in_2/y,out/x_2\}}\qquad AND(x_1,\neg y.x_2), OR(x_2,y_2,z).$$

$$x_1=\neg x^*, y_1=\neg y^*, x_2=\neg x\wedge y^*, \quad \frac{AND(x,\neg y,y_2),}{(31.5)\mid\{in_1/x,in_2/\neg y,out/y_2\}}\qquad OR(\neg x\wedge y,y_2,z).$$

$$x_1=\neg x^*, y_1=\neg y^*, x_2=\neg x\wedge y^*, y_2=x\wedge\neg y^*, \quad \frac{OR(\neg x\wedge y,x\wedge\neg y,z).}{(31.6)\mid\{in_1\;\;\neg x\wedge y, in_2/x\wedge\neg y,out/z\}}$$

$$x_1=\neg x^*, y_1=\neg y^*, x_2=\neg x\wedge y^*, y_2=x\wedge\neg y^*, z=(\neg x\wedge y)\vee(x\wedge\neg y)^*.$$

Fig. 31.6 First-level forward inference tree for the exclusive-or gate

In Fig. 31.6, the top formula is the structural formula, the bottom one is the pure behavioral formula, the formulas in between are a mixture. Every step of the inference is a unification of the leftmost structural subformula underlined with one of the antecedents of the transition formulas. If the unification is successful, then the consequent of the corresponding transition formula is inferred (it is a behavioral subformula). As many number of steps of inference is needed as the number of structural subformulas in the structural formula. Therefore, the computational complexity is linear. Each step of inference decreases one structural subformula, increases one behavioral subformula. Therefore, the algorithm is bound to terminate. The behavioral formula inferred at the bottom of Fig. 31.6 contains the behavioral subformula to be verified: $z=(\neg x\wedge y)\vee(x\wedge\neg y)$, therefore, the verification is a success, and the structural description Fig. 31.5 implements its behavioral description: $z=(\neg x\wedge y)\vee(x\wedge\neg y)$.

Now, let us analyze in detail the first step of inference in Fig. 31.6. NOT(x, x_1) is underlined, denoting that NOT(x, x_1) involves in the unification. On the left of the vertical line, there is the number (31.4), denoting that NOT(x, x_1) unifies with the antecedent of (31.4). On the right of the vertical line, there is $\{in/x, out/y\}$, denoting that in the unification, *in* is substituted by x, *out* by y. At the bottom of the vertical line, there is $x_1=\neg x^*$, denoting that after the unification, the consequent of (31.4): $x_1=\neg x^*$ is inferred (* denotes

that it is a behavioral subformula). After $x_l = \neg x^*$ is inferred, all x_l in the formula where the behavioral subformula is in are substituted by $\neg x$.

In (Bian, et.al; 2005), it take HOL 19 steps to verify the exclusive-or gate. While in this section we see that mutually-inversistic hardware verifier is automated.

31.2.3 Automated verification of sequential circuits

We use single pulser (Kropf, 1997) as an example to show the automated verification of the sequential circuits. The wave form diagram of single pulser is shown in Fig. 31.7.

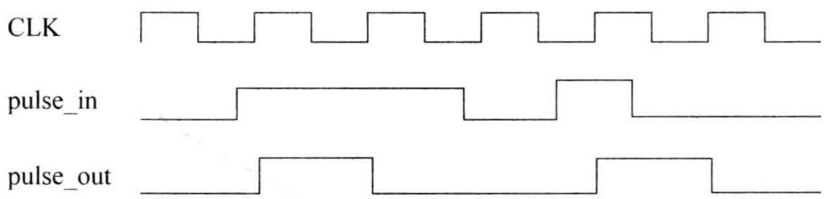


Fig. 31.7 Wave form diagram of single pulser

The working principle of single pulser is as follows: when the input (pulse_in) is in the high voltage level, and the rising edge of the next clock arrives, then the output (pulse_out) is in the high voltage level; however long the high voltage level of the input lasts, the high voltage level of the output lasts only one clock cycle.

The structural description of the single pulser is shown in Fig. 31.8, where DFF denotes D flip-flop.

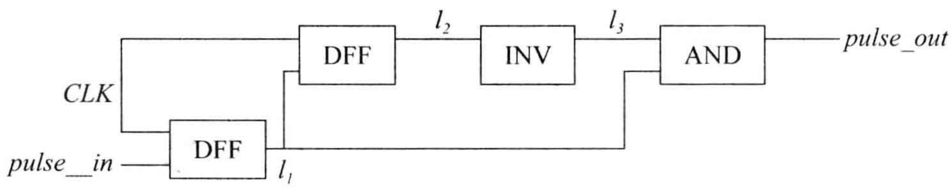


Fig. 31.8 Structural description of single pulser

The structural formula of single pulser is:

DFF(pulse_in, l_1), DFF(l_1, l_2), INV(l_2, l_3), AND(l_1, l_3, pulse_out). (31.8)

The transition formulas are:

DFF(D, Q)–: $Q(t+1)=D(t)^*$. (31.9)

INV(*in*, *out*)–: $out=\neg in^*$. (31.10)

AND(*in*$_1$, *in*$_2$, *out*)–: $out=in_1 \wedge in_2^*$. (31.11)

The behavioral subformula to be verified is:

pulse_out(t+1)=pulse_in(t)*, pulse_out(t+2)=$\neg$pulse_in(t+1)*. (31.12)

The first-level forward inference tree of single pulser is shown in Fig. 31.9.

Now, let us analyze Fig. 31.9. First, in the second step of inference, the system should have obtained $l_2(t+1)=l_1(t+0)$. But there is $l_1(t+1)$ in front, therefore, the system should also generates $l_2(t+2)=l_1(t+1)$. Merging the two, the system generates $l_2(t+[1, 2])=l_1(t+[0, 1])$. Secondly, $l_1(t+1)$=pulse_in(t) inferred in the first step should not be substituted into l_1 of the structural subformulas that follow, but it should wait until in the second step the behavioral subformula $l_2(t+[1, 2])=l_1(t+[0, 1])$ is inferred, then the system should substitute pulse_in(t+[−1, 0]) into $l_1(t+[0, 1])$. Thirdly, in the last step of inference, there are $l_1(t+[0, 1])$ and $l_3(t+[1, 2])$, therefore, the system should generates pulse_out(t+[0, 1, 2]). Lastly, the pure behavioral formula contains the behavioral subformula pulse_out(t+[0, 1, 2])=pulse_in(t+[−1, 0, 1])∧¬pulse_in(t+[−2, −1, 0]). The formula with the first indices in the square brackets is pulse_out(t+0)=pulse_in(t−1)∧¬pulse_in(t−2). In this formula, because pulse_in(t−1) is F, pulse_out(t+0) is F. The formula with the second indices in the square brackets is pulse_out(t+1)=pulse_in(t+0)∧¬pulse_in(t−1). In this formula, pulse_in(t−1) is F, ¬pulse_in(t−1) is T, and pulse_in(t+0) is T, therefore, pulse_out(t+1) is T. The formula with the third indices in the square brackets is pulse_out(t+2)=pulse_in(t+1)∧¬pulse_in(t+0). In this formula, pulse_in(t+0) is T, ¬pulse_in(t+0) is F, whatever pulse_in(t+1) is, pulse_out(t+2) remains F. To sum up, the structural description of single pulser implements its behavioral description.

DFF(pulse_in, l_1), DFF(l_1, l_2), INV(l_2, l_3), AND(l_1, l_3, pulse_out).

(31.9) | {pulse_in/D, l_1/Q}

l_1(t+1)=pulse_in(t)*, DFF(l_1, l_2,), INV(l_2, l_3), AND(l_1, l_3, pulse_out).

(31.9) | {l_1/D, l_2/Q}

l_1(t+1)=pulse_in(t)*, l_2(t+[1, 2])=l_1(t+[0, 1])*, INV(l_2, l_3), AND(l_1, l_3, pulse_out).

=pulse_in(t+[-1, 0])* (31.10) | {l_2/in, l_3/out}

l_1(t+1)=pulse_in(t)*, l_2(t=[1, 2])=l_1(t+[0, 1])*, l_3(t+[1, 2])=¬l_2(t+[1, 2])*,
AND(l_1, l_3, pulse_out).

=pulse_in(t+[−1, 0])* =¬pulse_in(t+[−1, 0])

(31.11) | {l_1/in_1, l_3/in_2, pulse_out/out}

l_1(t+1)=pulse_in(t)*, l_2(t+[1, 2])=l_1(t+[0, 1])*, l_3(t+[1, 2])=¬l_2(t+[1, 2])*, pulse_out(t+[0, 1, 2])
=l_1(t+[0, 1, 2])∧l_3(t+[0, 1, 2])*
=pulse_in(t+[-1, 0])* =¬pulse_in(t+[−1, 0])*, =pulse_in(t+[−1, 0, 1])∧¬pulse_in(t+[−2, −1, 0])*.

Fig. 31.9 First-level forward inference tree of single pulser

31.2.4 Automated verification of transistor-level circuits

The structural description of the CMOS inverter is shown in Fig. 31.10.

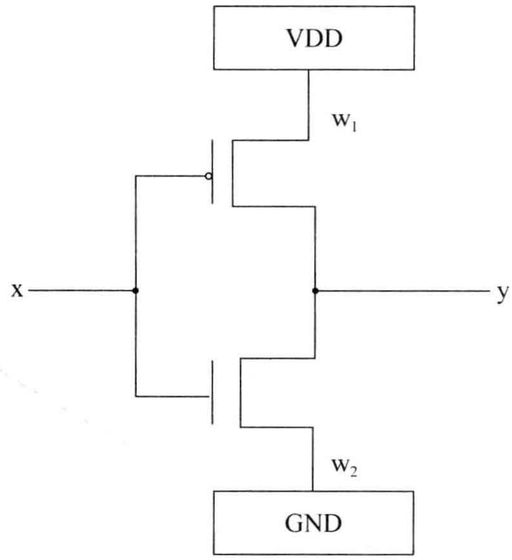


Fig. 31.10 Structural description of CMOS inverter

And we want to verify $y=\neg x$. The structural formula is:

$$\mathrm{VDD}(w_1),\ \mathrm{GND}(w_2),\ \mathrm{Ptran}(x, w_1, y),\ \mathrm{Ntran}(x, y, w_2). \tag{31.13}$$

The transition formulas are:

$$\mathrm{VDD}(u)\text{-:}\ u=\mathrm{T}^* \tag{31.14}$$

$$\mathrm{GND}(u)\text{-:}\ u=\mathrm{F}^* \tag{31.15}$$

$$\mathrm{Ptran}(u, v_1, v_2)\text{-:}\ \neg u\rightarrow(v_1=v_2)^* \tag{31.16}$$

$$\mathrm{Ntran}(u, v_1, v_2)\text{-:}\ u\rightarrow(v_1=v_2)^* \tag{31.17}$$

The behavioral subformula to be verified is:

$$\neg x\rightarrow(y=\mathrm{T})^*,\ x\rightarrow(y=\mathrm{F})^* \tag{31.18}$$

The first-level forward inference tree is shown in Fig. 31.11.

From the last row of Fig. 31.11 we see that the structural description implements its behavioral description. In (Kropf, 1999), 9 steps are taken to verify the CMOS inverter. While mutually-inversistic hardware verifier is automated in verifying the CMOS inverter.

$$\underline{\mathrm{VDD}(w_1)},\ \mathrm{GND}(w_2),\ \mathrm{Ptran}(x, w_1, y),\ \mathrm{Ntran}(x, y, w_2)$$

$$(31.14)\ \Big|\ \{u/w_1\}$$

$$w_1\text{=}\mathrm{T}^*,\ \underline{\mathrm{GND}(w_2)},\ \mathrm{Ptran}(x, \mathrm{T}, y).\ \mathrm{Ntran}(x, y, w_2)$$

$$(31.15)\ \Big|\ \{u/w_2\}$$

$$w_1\text{=}\mathrm{T}^*,\ w_2\text{=}\mathrm{F}^*,\ \underline{\mathrm{Ptran}(x, \mathrm{T}, y)},\ \mathrm{Ntran}(x, y, \mathrm{F})$$

$$(31.16)\ \Big|\ \{u/x,\ v_1/\mathrm{T},\ v_2/y\}$$

$$w_1\text{=}\mathrm{T}^*,\ w_2\text{=}\mathrm{F}^*, \neg x \rightarrow (\mathrm{T}\text{=}y)^*,\ \underline{\mathrm{Ntran}(x, y, \mathrm{F}).}$$

$$(31.17)\ \Big|\ \{u/x,\ v_1/y,\ v_2/\mathrm{F}\}$$

$$w_1\text{=}\mathrm{T}^*,\ w_2\text{=}\mathrm{F}^*, \neg x \rightarrow (\mathrm{T}\text{=}y)^*,\ x \rightarrow (y\text{=}\mathrm{F})^*$$

Fig. 31.11 First-level forward inference tree for CMOS inverter

Chapter 32 Axiomatic systems brought into mutually-inversistic automated decomposition systems

The axiomatic systems in this chapter are based on (Hamilton, 2003).

32.1 L system of propositional calculus of classical logic brought into third-level automated decomposition systems

L system of propositional calculus of classical logic has three axiom schemes:

(L_1) $A \to B \to A$.

(L_2) $(A \to B \to C) \to (A \to B) \to A \to C$.

(L_3) $(\neg A \to \neg B) \to B \to A$.

It also has an inference rule: MP: B can be inferred from $A \to B$ and A.

The proof of the theorem $A \to A$ is as follows:

$A \to (A \to A) \to A$	L_1	(32.1)
$\underbrace{(A \to (A \to A) \to A)}_{(1)} \to \underbrace{(A \to A \to A)}_{(2)} \to A \to A$	L_2	(32.2)
$(A \to A \to A) \to A \to A$	(32.1) (32.2) MP	(32.3)
$A \to A \to A$	L_1	(32.4)
$A \to A$	(32.3) (32.4) MP	(32.5)

Formula (32.2) is an instance of the axiom scheme L_2 where the subformulas marked with (1) and (2) are logical connection. Therefore, (32.2) is a connection of logical connections. MP takes the connection of logical connections (32.2) as the major premise, takes logical connection (32.1) as the minor premise, so, what it does is the third-level hypothetical inference.

An axiomatic system such as L system has only one inference rule MP. It can be brought into mutually-inversistic forward automated decomposition systems. Its equivalent third-level Prolog program and goal are as follows:

(L_1) $A \leftarrow B \leftarrow A$.

(L_2) $C \leftarrow A \leftarrow (B \leftarrow A) \leftarrow (C \leftarrow B \leftarrow A)$.

Goal: ?-$P \leftarrow P$.

The third-level SLD tree is shown in Fig. 32.1.

$?{-}P \rightarrow P.$

(L$_2$) $\{C_1/P, A_1/P\}$

$?{-}(B_1 \leftarrow P) \leftarrow (P \leftarrow B_1 \leftarrow P).$

(L$_1$) $\{B_1/A_2 \leftarrow B_2, A_1/\mathrm{P}\}$

$?{-}P \leftarrow (P \leftarrow B_2) \leftarrow P.$

(L$_1$) $\{A_3/P, B_3/P \leftarrow B_2\}$

$\square$

Fig. 32.1 Third-level SLD tree for $P \rightarrow P$

In Fig. 32.1, B_2 is not bound to a specific atomic proposition, so, it can be assigned either P or Q.

32.2 K$_{LG}$ system of predicate calculus with equality brought into second-level automated decomposition systems

Adding three axioms concerning quantifiers into L system, we obtain K$_L$ system of predicate calculus of classical logic:

(K$_1$) $A \rightarrow B \rightarrow A$

(K$_2$) $(A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow B) \rightarrow A \rightarrow C$

(K$_3$) $(\neg A \rightarrow \neg B) \rightarrow B \rightarrow A$

(K$_4$) $(\forall_x)A \rightarrow A$

(K$_5$) $(\forall_x)A(x) \rightarrow A(\mathrm{t})$

(K$_6$) $(\forall_x)(A \rightarrow B) \rightarrow A \rightarrow (\forall_x)B$

Adding three axioms (E$_7$) through (E$_9$) concerning the equality sign A_1^2; i.e., =, we obtain K$_{LG}$ system of predicate calculus with equality:

(E$_7$) $\mathrm{A}_1^2(x_l, x_l)$

(E$_8$) $\mathrm{A}_1^2(t_k, u) \rightarrow \mathrm{A}_1^2(f_i^n(t_l, \ldots, t_k, \ldots, t_n), f_i^n(t_l, \ldots, u, \ldots, t_n))$ where $t_l, \ldots, t_n, u$ are any terms, f_i^n is any function.

(E$_9$) $\mathrm{A}_1^2(t_k, u) \rightarrow A_i^n(t_l, \ldots, t_k, \ldots, t_n) \rightarrow A_i^n(t_l, \ldots, u, \ldots, t_n)$ where $t_l, \ldots, t_n, u$ are any terms, A_i^n is any predicate.

From Fig. 32.1, we see that B_l can be bound to $A_2 \leftarrow B_2$, an instance of the axiom scheme L$_2$ can be (32.2) (the connection of logical connections), therefore MP in L system makes third-level hypothetical inference. But, the propositional variables A, B, and C in the axiom schemes (K$_1$) through (K$_6$) of K$_{LG}$ system can only be assigned atomic propositions,

therefore, MP in K_{LG} system makes second-level hypothetical inference. K_{LG} system can be brought into second-level automated decomposition systems. The second-level Prolog program and goal for proving the mathematical theorem $A_1^2(x_1, x_2) \rightarrow A_1^2(x_2, x_1)$ are as follows:

(E$_7$) $A_1^2(u_1, u_1)$.

(E$_9$) $A_1^2(u_2, u_1) \leftarrow A_1^2(u_1, u_1) \leftarrow A_1^2(u_1, u_2)$.

(K$_1$) $A \leftarrow B \leftarrow A$.

(K$_2$) $C \leftarrow A \leftarrow (B \leftarrow A) \leftarrow (C \leftarrow B \leftarrow A)$.

Goal: $?\text{-}A_1^2(x_2, x_1) \leftarrow A_1^2(x_1, x_2)$.

The second-level SLD tree of the program and goal is shown in Fig. 32.2.

$$? - A_1^2(x_2, x_1) \leftarrow A_1^2(x_1, x_2).$$

(K$_2$) $\{C_3/A_1^2(x_2, x_1), A_3/A_1^2(x_1, x_2)\}$

$$? - B_3 \leftarrow A_1^2(x_1, x_2) \leftarrow (A_1^2(x_2, x_1) \leftarrow B_3 \leftarrow A_1^2(x_1, x_2)).$$

(K$_1$) $\{B_3/A_4, B_4/A_1^2(x_1, x_2)\}$

$$? - A_4 \leftarrow (A_1^2(x_2, x_1) \leftarrow A_4 \leftarrow A_1^2(x_1, x_2)).$$

(E$_7$) $\{A_4/A_1^2(u_1, u_1)\}$

$$? - A_1^2(x_2, x_1) \leftarrow A_1^2(u_1, u_1) \leftarrow A_1^2(x_1, x_2)$$

(E$_9$) $\{u_1/x_1\}$

$$\square$$

Fig. 32.2 Second-level SLD tree for $A_1^2(x_1, x_2) \rightarrow A_1^2(x_2, x_1)$

32.3 G system of group theory brought into second-level automated decomposition systems

Adding three axioms of group theory into K_{LG} system, we obtain G system of group theory:

(G$_1$) $f_1^2(f_1^2(x_1, x_2), x_3) = f_1^2(x_1, f_1^2(x_2, x_3))$ (associativity)

(G$_2$) $f_1^2(a_1, x_1) = x_1$ (left identity)

(G$_3$) $f_1^2(f_1^1(x_1), x_1) = a_1$ (left inverse)

where f_1^2 is multiplication; i.e., $*$, f_1^1 is the inverse operation; i.e., -1. Like K_{LG} system, MP in G system makes second-level hypothetical inference. It can be brought into second-level automated decomposition system.

Suppose a_1 is the identity. The second-level Prolog program and goal for proving $f_1^2(a_1, f_1^2(a_1, a_1))=a_1$ are as follows:

(G_2) $(\forall_{x1})(f_1^2(a_1, x_1)=x_1)$.

(K'_5) $(f_1^2(a_1, a_1)=a_1) \leftarrow (\forall_{x1})(f_1^2(a_1, x_1)=x_1)$.

(K''_5) $(f_1^2(a_1, f_1^2(a_1, a_1))=f_1^2(a_1, a_1)) \leftarrow (\forall_{x1})(f_1^2(a_1, x_1)=x_1)$.

(E_9) $f_1^2(a_1, f_1^2(a_1, a_1))=a_1 \leftarrow f_1^2(a_1, f_1^2(a_1, a_1))=f_1^2(a_1, a_1) \leftarrow f_1^2(a_1, a_1)=a_1$.

Goal: $?\text{-}f_1^2(a_1, f_1^2(a_1, a_1))=a_1$.

The second-level SLD tree is shown in Fig. 32.3.

$? \text{-} f_1^2(a_1, f_1^2(a_1, a_1))=a_1.$

(E_9)

$? \text{-} f_1^2(a_1, f_1^2(a_1, a_1))=f_1^2(a_1, a_1) \leftarrow f_1^2(a_1, a_1)=a_1.$

(K''_5)

$? \text{-} (\forall x_1)(f_1^2(a_1, x_1)=x_1) \leftarrow f_1^2(a_1, a_1)=a_1.$

(G_2)

$? \text{-} f_1^2(a_1, a_1)=a_1.$

(K'_5)

$? \text{-} (\forall x_1)(f_1^2(a_1, x_1)=x_1).$

(G_2)

$\square$

Fig. 32.3 Second-level SLD tree for $f_1^2(a_1, f_1^2(a_1, a_1))=a_1$

Part 12
Applications of implicit inductive compositions

Implicit inductive compositions are applied to logical theorem provers. Quasi-logical theorem prover is based on first-level implicit inductive composition. Single, multiple, and quasi-transcendent logical theorem provers are based on second-level implicit inductive composition. Quasi-, single, multiple, semi-, and quasi-transcendent logical theorem provers are established on the basis of the main-auxiliary algebras of quasi-, single, multiple, semi-, and quasi-transcendent set theorems respectively.

Chapter 33 Applications of implicit inductive compositions

33.1 Quasi-logical theorem prover and quasi-transcendent logical theorem prover

33.1.1 Proof of the quasi-logical theorem with the empirical or mathematical connection operator being $=^{-1}$

After the lexical analysis, the number of fact propositional variables in the quasi-logical connection proposition to be proved is known. The proof algorithm is as follows:

(1) Consider the fact propositional variables and $\varnothing_0$, U_0. For every fact propositional variable, the set of minterms containing the fact propositional variable is generated. For $\varnothing_0$, the empty set is generated. For U_0, the universal set is generated.

(2) For the left hand side and right hand side of $=^{-1}$, make composition operations.

(3) If the set of minterms of the left hand side is mutually inversely equivalent to that of the right hand side, then the quasi-logical connection proposition to be proved is a quasi-logical theorem, otherwise, it is not.

Example 33.1: Prove $P \wedge U_0 =^{-1} P$ to be a quasi-logical theorem.

Proof: The proposition has only one fact propositional variable P. Therefore, for P, $\{P\}$ is generated, for U_0, $\{\neg P, P\}$ is generated. The proposition becomes

$$\{P\} \cap \{\neg P, P\} =^{-1} \{P\} \tag{33.1}$$

The result of the composition operation of the left hand side is $\{P\}$, that of the right hand side is also $\{P\}$, the left hand side is mutually inversely equivalent to the right hand side. Therefore, $P \wedge U_0 =^{-1} P$ is a quasi-logical theorem.

Q.E.D.

33.1.2 Proof of the quasi-logical theorem with the empirical or mathematical connection operator being $\leqslant^{-1}$

After the lexical analysis, the number of fact propositional variables in the quasi-logical connection proposition to be proved is known. The proof algorithm is as follows:

(1) Consider the fact propositional variable. For every fact propositional variable, the set of minterms containing the fact propositional variable is generated.

(2) For the left hand side and right hand side of $\leqslant^{-1}$, make composition operations.

(3) If the left hand side is not an empty set, the right hand side is not a universal set, and the left hand side is mutually inversely contained in the right hand side, then the quasi-logical connection proposition to be proved is a quasi-logical theorem, otherwise, it is not.

Example 33.2: Prove $P \wedge Q \leqslant^{-1} P$ to be a quasi-logical theorem.

Proof: The proposition has P and Q two fact propositional variables, therefore, for P, $\{P\bar{Q}, PQ\}$ is generated; for Q, $\{\bar{P}Q, PQ\}$ is generated. The proposition becomes

$$\{P\bar{Q}, PQ\} \cap \{\bar{P}Q, PQ\} \subseteq^{-1} \{P\bar{Q}, PQ\} \tag{33.2}$$

The result of the composition operation of the left hand side is $\{PQ\}$, that of the right hand side is $\{P\bar{Q}, PQ\}$, the left hand side is not an empty set, the right hand side is not a universal set, the left hand side is mutually inversely contained in the right hand side, therefore, $P \wedge Q \leqslant^{-1} P$ is a quasi-logical theorem.

Q.E.D.

33.1.3 Quasi-transcendent logical theorem prover

The proof method of quasi-transcendent logical theorem is the same as that of quasi-logical theorem. Quasi-transcendent logical connection propositions are obtained by lifting P, Q, R, $\varnothing_0$, and U_0 quasi-logical connection propositions to Ψ, Ω, Θ, $\varnothing_1$, and U_1.

33.2 Single logical theorem prover

The algorithm of single logical theorem prover is as follows:

(1) If every fact propositional variable occurs at least twice, then the single logical connection proposition to be proved is bound, the proof process can go on; otherwise, the proposition to be proved is free, it is not a single logical theorem. For example, all of P, Q, and R in $\{P/\wedge^{-1}Q\} \wedge \{Q \leqslant^{-1} R\} \leqslant^{-1} \{P/\wedge^{-1} R\}$ occur twice, the proposition is bound, and the proof process can go on. While, R in $\{P \leqslant^{-1} Q\} \leqslant^{-1} \{P \wedge R \leqslant^{-1} Q\}$ occurs only once, the proposition is free, it is not a single logical theorem.

(2) Reduce the scope of $\neg$. $\neg\{P \leqslant^{-1} Q\}$ is changed to $P/\wedge^{-1} \neg Q$, $\neg\{P/\wedge^{-1} Q\}$ is changed to $P \leqslant^{-1} \neg Q$, $\neg\{P \vee /^{-1} Q\}$ is changed to $\neg P/\wedge^{-1} \neg Q$. The negation of a unicellular second-order single empirical or mathematical connection proposition is the disjunctions of the other six unicellular second-order single empirical or mathematical connection propositions. For example, $\neg\{P =^{-1} Q\}$ is changed to $\{P =^{-1} \neg Q\} \vee \{P \times^{-1} Q\} \vee \{P <^{-1} Q\} \vee \{P <^{-1} \neg Q\} \vee \{\neg P <^{-1} Q\} \vee \{\neg P <^{-1} \neg Q\}$.

(3) Change a unicellular second-order single empirical or mathematical connection proposition to multicellular second-order single empirical or mathematical

connection propositions. $P=^{-1}Q$ is changed to $\{P\leqslant^{-1}Q\}\wedge\{\neg P\leqslant^{-1}\neg Q\}$, $P<^{-1}Q$ is changed to $\{P\leqslant^{-1}Q\}\wedge\{\neg P/\wedge^{-1}Q\}$, $P\times^{-1}Q$ is changed to $\{P/\wedge^{-1}Q\}\wedge\{P/\wedge^{-1}\neg Q\}\wedge\{\neg P/\wedge^{-1}Q\}\wedge\{\neg P/\wedge^{-1}\neg Q\}$.

(4) Change the empirical or mathematical connection operator $\leqslant^{-1}$ to $\vee/^{-1}$. For example, change $P\leqslant^{-1}Q$ to $\neg P\vee/^{-1}Q$.

(5) After the lexical analysis, then we know the number of fact propositional variables in the single logical connection proposition to be proved. Change each fact propositional variable to the set of all the minterms containing the fact propositional variable. For example, change P in $\{\neg\{P\wedge\neg Q\}\vee/^{-1}R\}=^{-1}\{\neg P\vee/^{-1}Q\vee R\}$ to $\{\mathrm{PQR}, \mathrm{PQ\overline{R}}, \mathrm{P\overline{Q}R}, \mathrm{P\overline{QR}}\}$.

(6) Operate on fact composition operators. For example, $P\wedge Q$ is changed to $\{\mathrm{P\overline{Q}}, \mathrm{PQ}\}\cap\{\mathrm{\overline{P}Q}, \mathrm{PQ}\}$, its operation result is $\{\mathrm{PQ}\}$.

(7) Investigate whether the sets on both sides of every empirical or mathematical connection operator mutually inversely intercross. If so, then the subproposition formed by the empirical or mathematical connection operator connecting the left and right mutually inverse special propositions is a meaningful second-order single empirical or mathematical connection subpropositoin, and the proof process can go on, otherwise, it is a meaningless second-order single empirical or mathematical connection subproposition or quasi-logical subproposition, the proof process cannot go on. For example, $P/\wedge^{-1}Q$ in $\{P/\wedge^{-1}Q\}\wedge\{\neg Q\vee/^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is changed to $\{\mathrm{P\overline{QR}}, \mathrm{P\overline{Q}R}, \mathrm{PQ\overline{R}}, \mathrm{PQR}\}|\cap^{-1}\{\mathrm{\overline{P}Q\overline{R}}, \mathrm{\overline{P}QR}, \mathrm{PQ\overline{R}}, \mathrm{PQR}\}$.

The set on the left hand side and the set on the right hand side mutually inversely intercross: $\mathrm{PQ\overline{R}}$ and PQR belong to both the left hand side and the right hand side, $\mathrm{P\overline{QR}}$ and $\mathrm{P\overline{Q}R}$ belong to the left hand side but not the right hand side, $\mathrm{\overline{P}Q\overline{R}}$ and $\mathrm{\overline{P}QR}$ belong to the right hand side but not the left hand side, $\mathrm{\overline{PQR}}$ and $\mathrm{\overline{PQ}R}$ belong to neither the left hand side nor the right hand side. Therefore, $P/\wedge^{-1}Q$ is a meaningful single empirical or mathematical connection subproposition. For example, $P\wedge\neg P\vee/^{-1}Q$ in $\{P\wedge\neg P\vee/^{-1}Q\}\leqslant^{-1}\{P\vee/^{-1}Q\}$ is changed to

$$\{\mathrm{P\overline{Q}}, \mathrm{PQ}\}\cap\{\mathrm{\overline{PQ}}, \mathrm{\overline{P}Q}\}\cup|^{-1}\{\mathrm{\overline{P}Q}, \mathrm{PQ}\}.$$

Operate on $\cap$, obtaining

$$\{\}\cup|^{-1}\{\mathrm{\overline{P}Q}, \mathrm{PQ}\}.$$

The set on the left hand side and the set on the right hand side do not mutually inversely intercross: no minterm belongs to both the left hand side and the right hand side, no minterm belongs to the left hand side but not the right hand side. Therefore, $P\wedge\neg P\vee/^{-1}Q$ is a meaningless second-order single empirical or mathematical connection subproposition, and $\{P\wedge\neg P\vee/^{-1}Q\}\leqslant^{-1}\{P\vee/^{-1}Q\}$ is not a single logical theorem. For example, for $P\wedge Q\leqslant^{-1}P$ in $\{P\wedge Q\leqslant^{-1}P\}\leqslant^{-1}\{P\leqslant^{-1}Q\}$,

change the empirical or mathematical connection operator $\leqslant^{-1}$ to $\vee/^{-1}$, obtaining $\neg\{P\wedge Q\}\vee/^{-1}P$, change it to the set of minterms form, obtaining

$$\sim\{\{\mathrm{P\overline{Q}, PQ}\}\cap\{\mathrm{\overline{P}Q, PQ}\}\}\cup|^{-1}\{\mathrm{P\overline{Q}, PQ}\}.$$

Operate on $\cap$, obtaining

$$\sim\{\mathrm{PQ}\}\cup|^{-1}\{\mathrm{P\overline{Q}, PQ}\}.$$

Operate on $\sim$, obtaining

$$\{\mathrm{\overline{PQ}, \overline{P}Q, P\overline{Q}}\}\cup|^{-1}\{\mathrm{P\overline{Q}, PQ}\}.$$

The set on the left hand side and the set on the right hand side do not mutually inversely intercross: no minterm belongs to neither the left hand side nor the right hand side. Therefore, $P\wedge Q\leqslant^{-1}P$ is a quasi-logical connection subproposition, and $\{P\wedge\neg P\vee/^{-1}Q\}\leqslant^{-1}\{P\vee/^{-1}Q\}$ is not a single logical theorem.

(8) In this step, the main-auxiliary algebras for unicellular single set theorems $\mathrm{US_n}$ will be used. $\mathrm{US_2}$, $\mathrm{US_3}$, and $\mathrm{US_4}$ are shown in Figs. 33.1 through 33.3 respectively, in which the concrete vertices are the real vertices, and the hollow vertices are the imaginary vertices. In Fig. 33.1, the set of real vertices $\{P<^{-1}Q, P=^{-1}Q\}$; i.e., $\{\{\mathrm{\overline{PQ}, \overline{P}Q, PQ}\}, \{\mathrm{\overline{PQ}, PQ}\}\}$ of the substructure with $P<^{-1}Q$ being the top vertex and $\varnothing$ being the bottom vertex is the set of real vertices without the minterm $\mathrm{P\overline{Q}}$. Likewise, we can obtain the sets of real vertices without the minterms $\mathrm{P\overline{Q}}$, $\mathrm{\overline{P}Q}$, and PQ. The set of real vertices $\{P=^{-1}Q, P<^{-1}Q, \neg P<^{-1}\neg Q, \neg P<^{-1}Q, P\times^{-1}Q\}$ of the substructure with $\{\mathrm{PQ}\}$ being the bottom vertex and $P\times^{-1}Q$; i.e., $\mathrm{S_2}$ being the top vertex is the set of real vertices with the minterm PQ. Likewise, we can obtain the

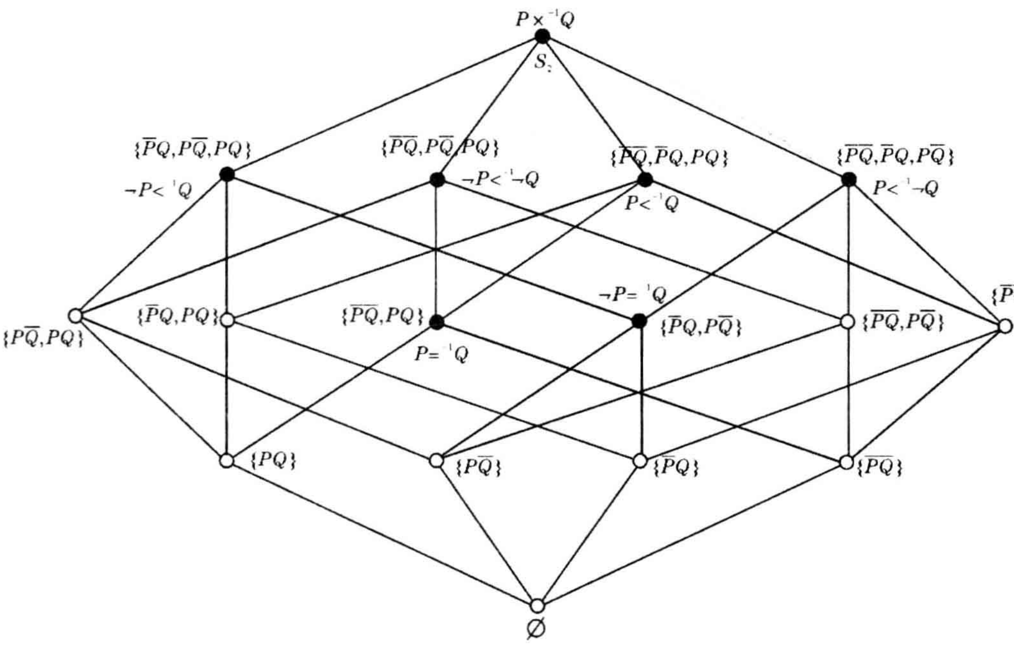


Fig. 33.1 Hasse diagram for $\mathrm{US_2}$

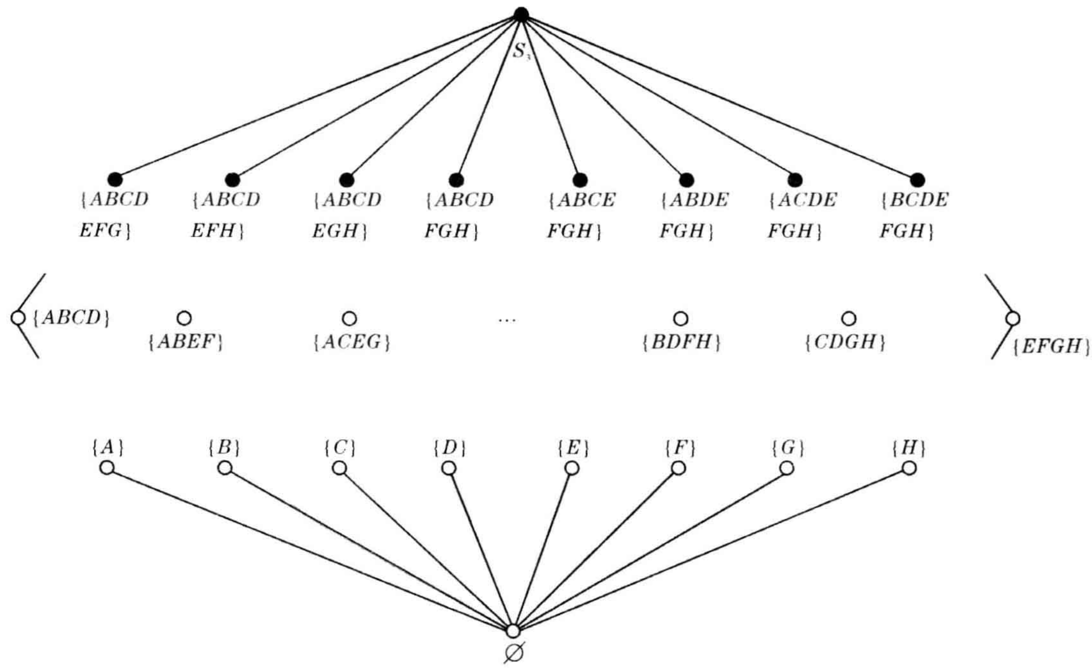


Fig. 33.2 Sketch of Hasse diagram for $\mathbf{US_3}$

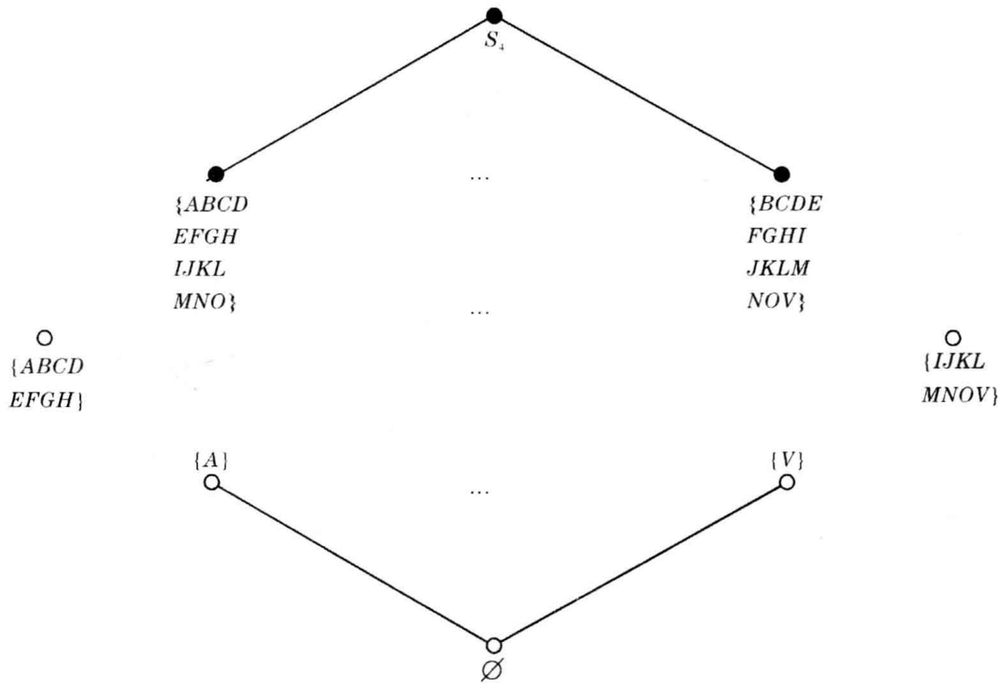


Fig. 33.3 Sketch of Hasse diagram for $\mathbf{US_4}$

sets of real vertices with the minterms $\overline{PQ}$, $\overline{P}Q$, and $P\overline{Q}$. In Fig. 33.2, A, B, …,G and H denote the minterms PQR, $PQ\overline{R}$, …, $\overline{PQ}R$, and $\overline{PQR}$ respectively. The set of real

vertices of the substructure with {A, B, C, D, E, F, G} being the top vertex and ∅ being the bottom vertex is the set of real vertices without the minterm H; i.e., $\overline{PQR}$. Likewise, we can obtain the sets of real vertices without the other minterms. The set of real vertices of the substructure with {A} being the bottom vertex and S_3 being the top vertex is the set of real vertices with the minterm A; i.e., PQR. Likewise, we can obtain the sets of real vertices with the other minterms. Similar discussion can be held for Fig. 33.3.

This step is to find the set of real vertices of a meaningful second-order single empirical or mathematical connection subproposition. The set of real vertices of $X|\cap^{-1}Y$ is the set of real vertices of the minterms that are contained in both X and Y. If there are more than one such minterms, then their union set should be computed. For example, the set of minterms of $P/\wedge^{-1}Q$ in $\{P/\wedge^{-1}Q\}\wedge\{\neg Q\vee/^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is

$$\{\mathrm{P\overline{QR}, P\overline{Q}R, PQ\overline{R}, PQR}\}|\cap^{-1}\{\mathrm{\overline{P}Q\overline{R}, \overline{P}QR, PQ\overline{R}, PQR}\}.$$

The minterms that are contained in both the left hand side and the right hand side are $\mathrm{PQ\overline{R}}$ and PQR. Thus, the set of real vertices of $P/\wedge^{-1}Q$ is the union set of the set of real vertices containing the minterm $\mathrm{PQ\overline{R}}$ and the set of real vertices containing the minterm PQR. The set of real vertices of the meaningful second-order single empirical or mathematical connection subproposition $X\cup|^{-1}Y$ is the set of vertices of the minterms that are contained in neither X nor Y. If there are more than one such minterms, then their intersection set should be computed. For example, the set of minterms of $\neg Q\vee/^{-1}R$ in $\{P/\wedge^{-1}Q\}\wedge\{\neg Q\vee/^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$ is

$$\{\mathrm{\overline{PQR}, \overline{PQ}R, P\overline{QP}, P\overline{Q}R}\}\cup|^{-1}\{\mathrm{\overline{PQR}, \overline{P}QR, P\overline{Q}R, PQR}\}.$$

The minterms that are contained in neither the left hand side nor the right hand side are $\mathrm{\overline{P}Q\overline{R}}$ and $\mathrm{PQ\overline{R}}$. Thus, the set of real vertices of $\neg Q\vee/^{-1}R$ is the intersection set of the set of real vertices without the minterm $\mathrm{\overline{P}Q\overline{R}}$ and the set of real vertices without the minterm $\mathrm{PQ\overline{R}}$.

(9) Operate on empirical or mathematical composition operators. For example, in $\{P/\wedge^{-1}Q\}\wedge\{\neg Q\vee/^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$, make $\cap$ operation on the sets of real vertices of $P/\wedge^{-1}Q$ and $\neg Q\vee/^{-1}R$.

(10) Change the logical connection operators $\vee/^{-1}$ and $/\wedge^{-1}$ to $\leqslant^{-1}$. For example, change $\{P/\wedge^{-1}Q\}\vee/^{-1}\{P/\wedge^{-1}\neg Q\}$ to $\neg\{P/\wedge^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}\neg Q\}$. Thus, only three kind of logical connection operators are left: $\leqslant^{-1}$, $<^{-1}$, and $=^{-1}$.

(11) If the set on the left hand side of the logical connection operator is not an empty set, the set on the right is not a universal set, then the single logical connection proposition to be proved is meaningful, and the proof process can go on;

otherwise, it is meaningless, and the proof process cannot go on.

(12) Determine whether the meaningful single logical connection proposition is a single logical theorem or not. For the logical connection operator $=^{-1}$, if the set on the left hand side is mutually inversely equivalent to the set on the right hand side, then the proposition to be proved is a single logical theorem; otherwise, it is not. For the logical connection operator $<^{-1}$, if the set on the left is mutually inversely properly contained in the set on the right, then the proposition to be proved is a single logical theorem; otherwise, it is not. For the logical connection operator $\leqslant^{-1}$, if the set on the left is mutually inversely contained in the set on the right, then the proposition to be proved is a single logical theorem; otherwise, it is not.

Example 33.3: Prove $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ to be a single logical theorem.

Proof: According to step (4), change the proposition to be proved to $\{\neg P\vee/^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$. According to step (5), change the result of the previous step to

$$\{\sim\{P\bar{Q}, PQ\}\cup|^{-1}\{\bar{P}Q, PQ\}\}\subseteq^{-1}\{\{P\bar{Q}, PQ\}|\cap^{-1}\{\bar{P}Q, PQ\}\}.$$

According to step (6), change the result of the previous step to

$$\{\{\overline{PQ}, \bar{P}Q\}\cup|^{-1}\{\bar{P}Q, PQ\}\}\subseteq^{-1}\{\{P\bar{Q}, PQ\}|\cap^{-1}\{\bar{P}Q, PQ\}\}.$$

According to step (7), investigate whether the sets on both sides of the empirical or mathematical connection operators $\cup|^{-1}$ and $|\cap^{-1}$ mutually inversely intercross. For $\{\overline{PQ}, \bar{P}Q\}\cup|^{-1}\{\bar{P}Q, PQ\}$, the set on the left $\{\overline{PQ}, \bar{P}Q\}$ and the set on the right $\{\bar{P}Q, PQ\}$ mutually inversely intercross: $\bar{P}Q$ belongs to both the left and the right, $\overline{PQ}$ belongs to the left but not the right, PQ belongs to the right but not the left, $P\bar{Q}$ belongs to neither the left nor the right. Therefore, $\neg P\vee/^{-1}Q$ is a meaningful second-order single empirical or mathematical connection subproposition. For the same reason, the set on the left and right of $\{P\bar{Q}, PQ\}|\cap^{-1}\{\bar{P}Q, PQ\}$ mutually inversely intercross, and $P/\wedge^{-1}Q$ is meaningful. According to step (8), $\{\overline{PQ}, \bar{P}Q\}\cup|^{-1}\{\bar{P}Q, PQ\}$ is without the minterm $P\bar{Q}$, $\{P\bar{Q}, PQ\}|\cap^{-1}\{\bar{P}Q, PQ\}$ is with the minterm PQ, we obtain

"the set of real vertices without the minterm $P\bar{Q}$" $\subseteq^{-1}$ "the set of real vertices with the minterm PQ".

That is,

$$\{P<^{-1}Q, P=^{-1}Q\}\subseteq^{-1}\{P=^{-1}Q, P<^{-1}Q, \neg P<^{-1}\neg Q, \neg P<^{-1}Q, P\times^{-1}Q\}.$$

According to step (10), $\{P<^{-1}Q, P=^{-1}Q\}$ is not an empty set, and $\{P=^{-1}Q, P<^{-1}Q, \neg P<^{-1}\neg Q, \neg P<^{-1}Q, P\times^{-1}Q\}$ is not a universal set, therefore, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is meaningful. According to step (12), $\{P<^{-1}Q, P=^{-1}Q\}$ is mutually inversely contained in $\{P=^{-1}Q, P<^{-1}Q, \neg P<^{-1}\neg Q, \neg P<^{-1}Q, P\times^{-1}Q\}$, therefore, $\{P\leqslant^{-1}Q\}\leqslant^{-1}\{P/\wedge^{-1}Q\}$ is a single logical theorem.

Q.E.D.

33.3 Multiple logical theorem prover

33.3.1 Chaining representation of the main-auxiliary algebra for multiple logical theorem M_3

A vertex of M_3 is shown in Fig. 33.4.

Pointer pointing to the first lower proposition	Pointer pointing to the main complement	Pointer pointing to the second lower proposition
Main property proposition segment	Equivalent property proposition segment	

Fig. 33.4 Vertex in M_3

The chaining representation of M_3 is shown in Fig. 33.5. In Fig. 33.5, NIL denotes that the value or pointer does not exist; vertex 1 is the starting vertex, vertex 6 is the ending vertex; vertices 1, 2, 4, and 6 constitute the left route, vertices 1, 3, 5, and 6 constitute the right route.

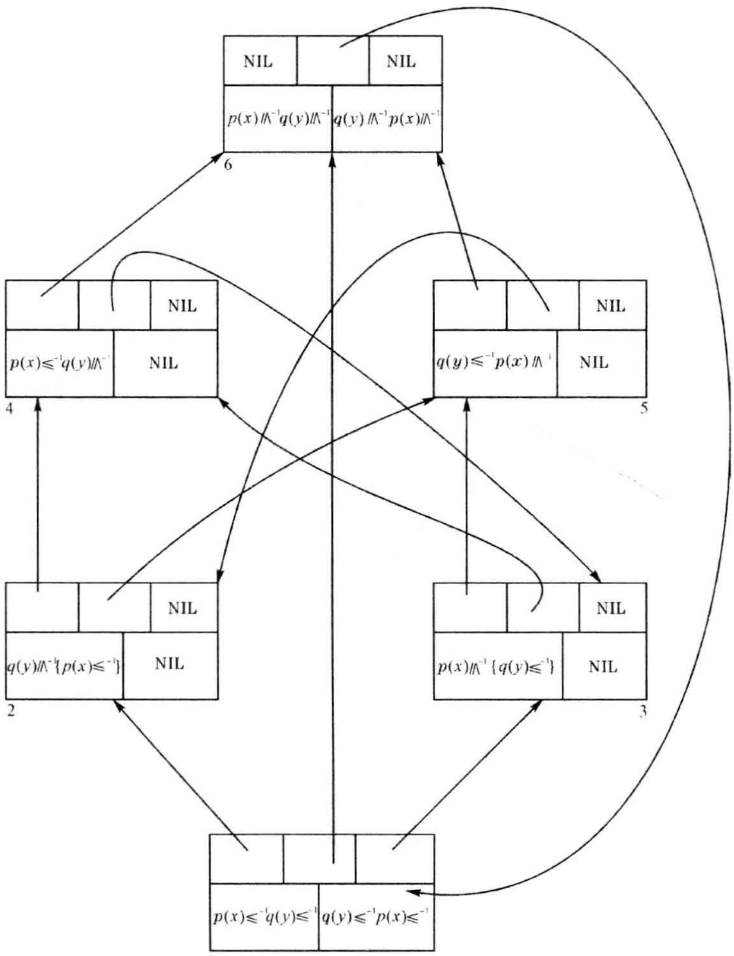


Fig. 33.5 Chaining representation of M_3

33.3.2 Algorithm of multiple logical theorem prover with three fact propositions

We call the subproposition on the left hand side of the logical connection operator of the proposition to be proved the left mutually inverse special proposition, in which the property proposition segment is the left property proposition segment, the nonproperty proposition is the left nonproperty proposition; the subproposition on the right hand side of the logical connection operator the right mutually inverse special proposition, in which the property proposition segment is the right property proposition segment, the nonproperty proposition is the right nonproperty proposition.

The proof algorithm is as follows:

(1) Along the left route of Fig. 33.5, from the starting vertex to the ending vertex, search for the left property proposition segment. If not found, then search along the right route, until the left property proposition segment is found. When searching, first search the main property proposition segment, then search the equivalent property proposition segment.

(2) If the left nonproperty proposition is the same as the right one, then the proposition to be proved can only be the subalternation proposition or the equivalence proposition, and continue; otherwise, it can only be the contradictory proposition, contrary proposition, or subcontrary proposition, and jump to step (7).

(3) Along the route where the left property proposition segment situates, search for the right property proposition segment; if found, then continue; otherwise, the proposition to be proved is not a multiple logical theorem.

(4) If the right property proposition segment is the upper proposition of the left property proposition segment, then the proposition to be proved is not a multiple logical theorem; otherwise, continue.

(5) If the right property proposition segment is the corresponding proposition of the left property proposition segment; i.e., they are the same vertices, then generate an equivalence proposition and compare it with the proposition to be proved; if they are the same, then the proposition to be proved is a multiple logical theorem in the form of equivalence proposition. If the right property proposition segment is not the corresponding proposition of the left one, then continue.

(6) If the right property proposition segment is the lower proposition of the left property proposition segment, then generate subalternation propositions and compare them with the proposition to be proved. If one subalternation proposition is the same as the proposition to be proved, then the proposition to be proved is a multiple logical theorem; otherwise, it is not.

(7) Find the main complement of the left property proposition segment.

(8) Along the different route of the route where the left property proposition segment situates, from the starting vertex to the ending vertex, search for the right property proposition segment. When searching, first search the main property proposition segment. If not found, then search the equivalent property proposition segment. If found, then continue; otherwise, the proposition to be proved is not a multiple logical theorem.

(9) If the right property proposition segment is the upper proposition of the main complement of the left property proposition segment, then generate contrary propositions and compare them with the proposition to be proved. If one contrary proposition is the same as the proposition to be proved, then the proposition to be proved is a multiple logical theorem in the form of contrary proposition; otherwise, it is not a multiple logical theorem. If the right property proposition segment is not the upper proposition of the main complement of the left one, then continue.

(10) If the right property proposition segment is the main complement of the left property proposition segment, then generate contradictory propositions and compare them with the proposition to be proved. If one contradictory proposition is the same as the proposition to be proved, then the proposition to be proved is a multiple logical theorem in the form of contradictory proposition; otherwise it is not a multiple logical theorem. If the right property proposition segment is not the main complement of the left one, then continue.

(11) If the right property proposition segment is the lower proposition of the main complement of the left property proposition segment, then generate subcontrary propositions and compare them with the proposition to be proved. If one subcontrary proposition is the same as the proposition to be proved, then the proposition to be proved is a multiple logical theorem in the form of subcontrary proposition; otherwise, it is not a multiple logical theorem.

33.3.3 Examples

Example 33.4: Prove that $\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}r(x, y)\}\}=^{-1}\{p(x)\leqslant^{-1}q(y)/\wedge^{-1}r(x, y)\}$ is not a multiple logical theorem.

Proof: According to step (1), the left property proposition segment $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}$ is found at vertex 2 of the left route. According to step (2), the left nonproperty proposition is the same as the right one, both are $r(x, y)$. According to step (3), the right property proposition segment $p(x)\leqslant^{-1}q(y)/\wedge^{-1}$ is found at vertex 4 of the left route. According to step (6), the right property proposition segment $p(x)\leqslant^{-1}q(y)/\wedge^{-1}$ is the lower proposition of the left property proposition segment $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}$, and generate the subalternation proposition $\{q(y)/\wedge^{-1}$

$\{p(x)\leqslant^{-1} r(x, y)\}\}\leqslant^{-1}\{p(x)\leqslant^{-1} q(y)/\wedge^{-1} r(x, y)\}$, which is not the same as the proposition to be proved. Therefore, the proposition to be proved is not a multiple logical theorem.

Q.E.D.

Example 33.5: Prove that $\neg\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1} r(x, y)\}\}\vee/^{-1}\neg\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1} \neg r(x, y)\}\}$ is a multiple logical theorem.

Proof: According to step (1), the left property proposition segment $q(y)/\wedge^{-1}\{p(x)\leqslant^{-1}$ is found at vertex 2 of the left route. According to step (2), the left nonproperty proposition is $r(x, y)$, the right nonproperty proposition is $\neg r(x, y)$, they are not the same. According to step (7), the main complement $q(y)\leqslant^{-1} p(x)/\wedge^{-1}$ of the left property proposition segment is found at vertex 5 of the right route. According to step (8), the right property proposition segment $p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}$ is found at vertex 3 of the right route. According to step (9), the right property proposition segment $p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}$ is an upper proposition of the main complement $q(y)\leqslant^{-1} p(x)/\wedge^{-1}$ of the left property proposition segment. A contrary proposition $\neg\{\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1} r(x, y)\}\}/\wedge^{-1}\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}\neg r(x, y)\}\}\}$ is generated, it is not the same as the proposition to be proved. Then another contrary proposition $\neg\{q(y)/\wedge^{-1}\{p(x)\leqslant^{-1} r(x, y)\}\}\vee/^{-1}\neg\{p(x)/\wedge^{-1}\{q(y)\leqslant^{-1}\neg r(x, y)\}\}$ is generated, it is the same as the proposition to be proved. Therefore, the proposition to be proved is a multiple logical theorem.

Q.E.D.

Note: in Fig. 33.5, the upper proposition of a proposition is beneath the proposition in question, the lower proposition of a proposition is above the proposition in question.

33.4 Semilogical theorem prover

The semilogical connection proposition to be proved is in the form of:

"The first properly implicational single empirical or mathematical connection proposition to be proved"$\wedge$"the second properly implicational single empirical or mathematical connection proposition to be proved"$=^{-1}$ "the fact proposition to be proved". The chaining representation of the main-auxiliary algebra for semiset theorems SE_2 is shown in Fig. 33.6.

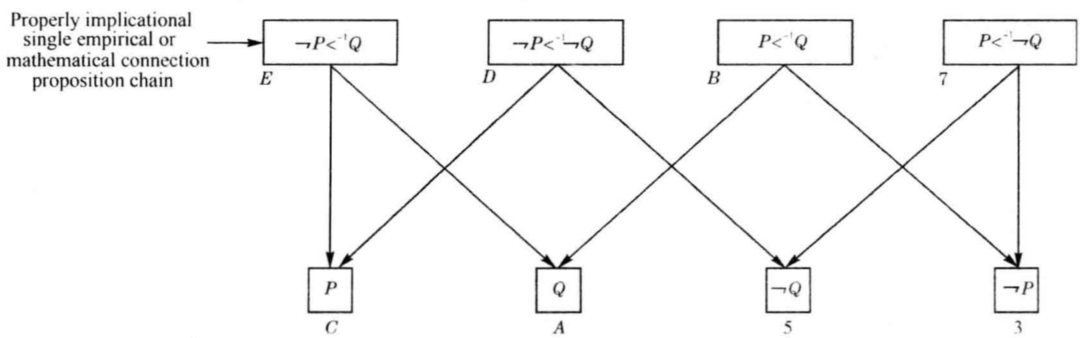


Fig. 33.6 Chaining representation of the main-auxiliary algebra for semiset theorems SE_2

The algorithm of the semilogical theorem prover is as follows:

(1) Along the properly implicational empirical or mathematical connection proposition chain of Fig. 33.6 search for the first properly implicational empirical or mathematical connection proposition to be proved. If not found, then the semilogical connection proposition to be proved is not a semilogical theorem; otherwise, continue.

(2) Along the properly implicational empirical or mathematical connection proposition chain of Fig. 33.6 search for the second properly implicational empirical or mathematical connection proposition to be proved. If not found, then the semilogical connection proposition to be proved is not a semilogical theorem; otherwise, continue.

(3) Investigate whether the pointers of the two propositions point to the same fact proposition. If not, then the semilogical connection proposition to be proved is not a semilogical theorem; otherwise, continue.

(4) Investigate whether the fact proposition pointed to is the same as the fact proposition to be proved. If not, then the semilogical connection proposition to be proved is not a semilogical theorem; otherwise, it is.

Example 33.6: Prove $\{\neg P<^{-1}Q\}\wedge\{\neg P<^{-1}\neg Q\}=^{-1}P$ to be a semilogical theorem.

Proof: According to step (1), $\neg P<^{-1}Q$ is found at vertex E of the properly implicational empirical or mathematical connection proposition chain. According to step (2), $\neg P<^{-1}\neg Q$ is found at vertex D of the properly implicational empirical or mathematical connection proposition chain. According to step (3), the two propositions point to the fact proposition P of vertex C. According to step (4), the fact proposition P pointed to is the same as the fact proposition P to be proved. Therefore, the semilogical connection proposition to be proved is a semilogical theorem.

Q.E.D.

Part 13
Applications of explicit inductive composition

In this part, mutually-inversistic machine learning, multiple connection operators association rule mining, and mutually-inversistic program refinement are introduced.

Chapter 34 Mutually-inversistic machine learning

34.1 Introduction to mutually-inversistic machine learning

In hypothetical inference, the major premise and minor premise are known, the conclusion is sought. For examples, the major premise $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ and the minor premise $\{\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)\}\wedge\{\mathrm{rat}(x)\leqslant^{-1}\mathrm{real}(x)\}$ are known, the conclusion $\mathrm{int}(x)\leqslant^{-1}\mathrm{real}(x)$ is sought. Mutually-inversistic machine learning is the inverse operation of hypothetical inference. There are two kind of mutually-inversistic machine learning. One is that the major premise and the conclusion are known, the minor premise is sought. For example, the major premise $\{P\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P\leqslant^{-1}R\}$ and the conclusion $\mathrm{int}(x)\leqslant^{-1}\mathrm{real}(x)$ are known, the minor premise $\{\mathrm{int}(x)\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}\mathrm{real}(x)\}$ is sought. This kind of machine learning is called inverse hypothetical inference towards the minor premise. The other is that the minor premise and the conclusion are known, the major premise is sought. For example, the minor premise $\{\mathrm{int}(x)\leqslant^{-1}\mathrm{rat}(x)\}\wedge\{\mathrm{rat}(x)\leqslant^{-1}\mathrm{real}(x)\}$ and the conclusion $\mathrm{int}(x)\leqslant^{-1}\mathrm{real}(x)$ are known, the major premise $\neg\{P\leqslant^{-1}Q\}\vee\neg\{Q\leqslant^{-1}R\}\vee/^{-1}\{P\leqslant^{-1}R\}$ is sought. This kind of machine learning is called inverse hypothetical inference towards the major premise. There are 16 kind of mutually-invresistic machine learning systems according as whether they are towards the minor premise or the major premise, whether they are first-level or second-level systems, whether they are single quasi-systems or multiple systems, whether they are decomposition systems or expert systems. The classification of these systems is shown in Fig. 34.1.

34.2 Mutually-inversistic machine learning systems

34.2.1 Inverse first-level single quasi-decomposition system towards the major premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.2.

inverse second-level single quasi decomposition system towards major premise	inverse second-level single quasi decomposition system towards major premise	inverse second-level multiple decomposition system towards major premise	inverse second-level multiple decomposition system towards major premise	decomposition system	second level
inverse second-level single quasi expert system towards major premise	inverse second-level single quasi expert system towards major premise	inverse second-level multiple expert system towards major premise	inverse second-level multiple expert system towards major premise	expert system	
inverse first-level single quasi decomposition system towards major premise	inverse first-level single quasi decomposition system towards major premise	inverse first-level multiple decomposition system towards major premise	inverse first-level multiple decomposition system towards major premise	decomposition system	first level
inverse first-level single quasi expert system towards minor premise	inverse first-level single quasi expert system towards minor premise	inverse first-level multiple expert system towards minor premise	inverse first-level multiple expert system towards minor premise	expert system	
minor premise	minor premise	minor premise	minor premise		
single quasi		multiple			

Fig 34.1 Classification of mutually-inversistic machine learning systems

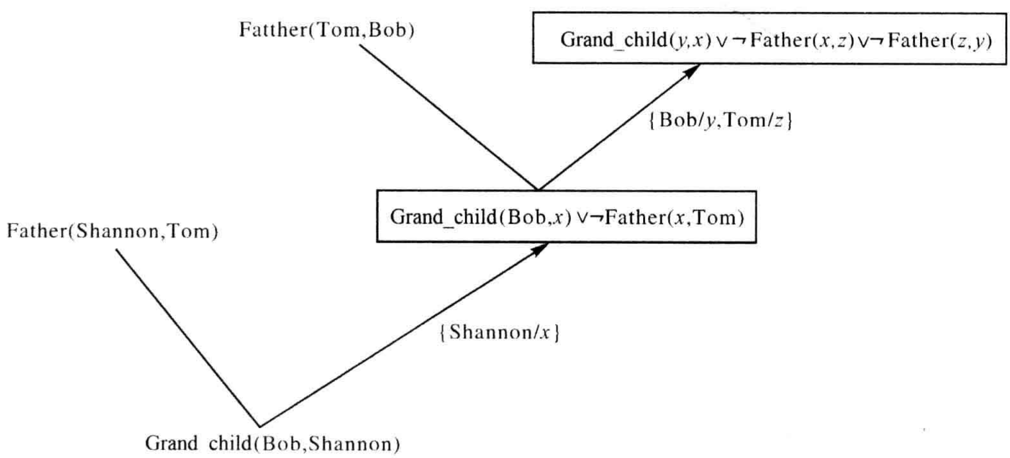


Fig. 34.2 Inverse first-level single quasi-decomposition system towards the major premise

FOIL in inductive logic programming also belongs to this kind of machine learning.

34.2.2 Inverse first-level single quasi-decomposition system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.3.

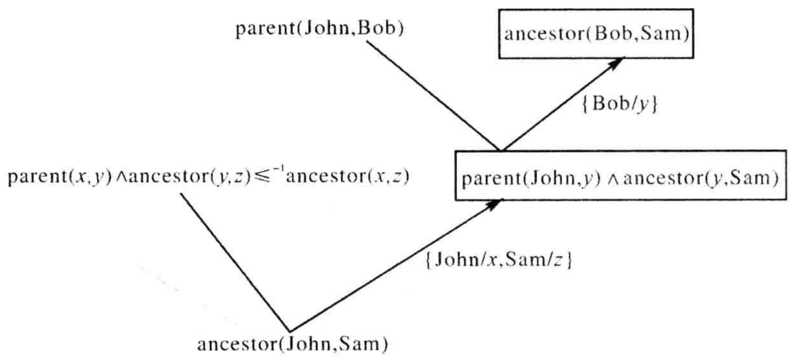


Fig.34.3 Inverse first-level single quasi-decomposition system towards the minor premise

34.2.3 Inverse first-level single quasi-expert system towards the major premise

Decision tree learning in inductive learning, rough set learning, and genetic algorithm belong to this kind of mutually-inversistic machine learning.

34.2.4 Inverse first-level single quasi-expert system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.4.

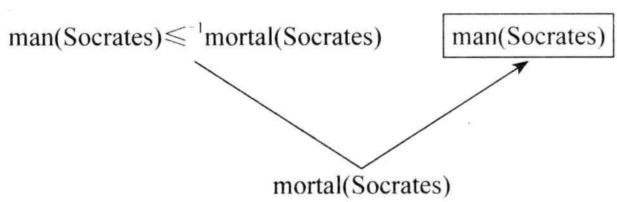


Fig. 34.4 Inverse first-level single quasi-expert system towards the minor premise

34.2.5 Inverse second-level single quasi-decomposition system towards the major premise

34.2.5.1 Second-level FOIL

Second-level FOIL is obtained by lifting FOIL one level up.

34.2.5.2 Inverse resolution

This kind of inverse resolution is obtained by lifting Fig. 34.2 one level up, shown in Fig. 34.5.

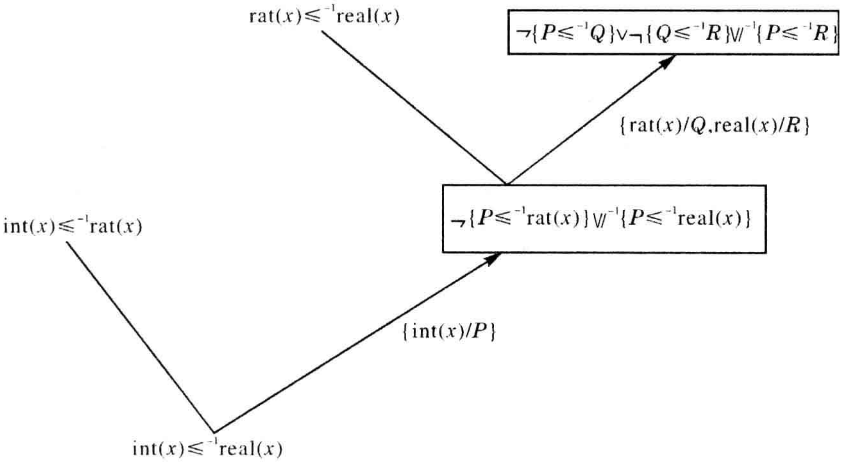


Fig. 34.5 Inverse second-level single quasi-decomposition system towards the major premise

34.2.6 Inverse second-level single quasi-decomposition system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.6.

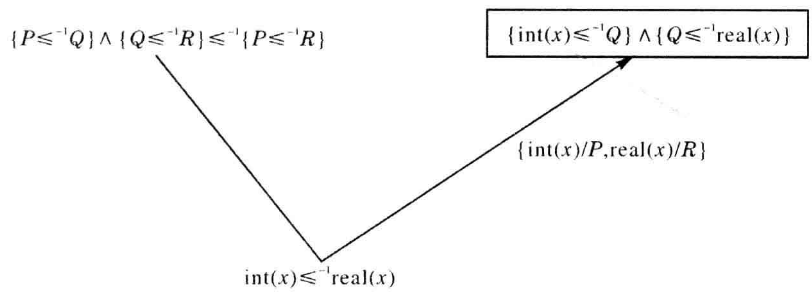


Fig. 34.6 Inverse second-level single quasi-decomposition system towards the minor premise

In the inductive logic programming, there is the inverse resolution shown in Fig. 34.7.

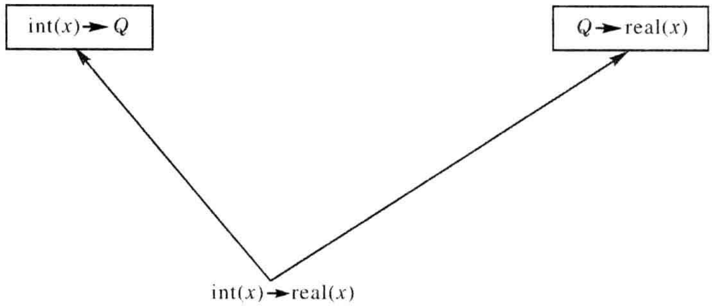


Fig. 34.7 Inverse resolution in inductive logic programming

Both Figs. 34.6 and 34.7 can obtain $\{\mathrm{int}(x)\leqslant^{-1}Q\}\wedge\{Q\leqslant^{-1}\mathrm{real}(x)\}$ $((\mathrm{int}(x)\rightarrow Q)\wedge(Q\rightarrow\mathrm{real}(x)\})$ from $\mathrm{int}(x)\leqslant^{-1}\mathrm{real}(x)$ $(\mathrm{int}(x)\rightarrow\mathrm{real}(x))$, therefore, we regard them as the same kind of machine learning. But actually they are different. Fig. 34.7 is the inverse resolution of the resolution shown in Fig. 34.8, while Fig. 34.6 is the inverse resolution of the resolution shown in Fig. 34.9.

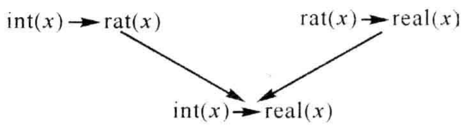


Fig. 34.8 Resolution in classical logic

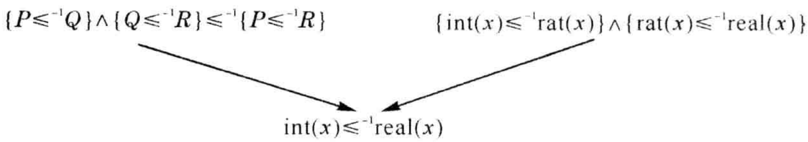


Fig. 34.9 Resolution in mutually-inversistic logic

In the inverse second-level single quasi-decomposition system towards the minor premise, $\{\mathrm{pos}(x)/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}\mathrm{int}(x)\}$ can also be learned from $\mathrm{pos}(x)/\wedge^{-1}\mathrm{int}(x)$ and $\{P/\wedge^{-1}Q\}\wedge\{Q\leqslant^{-1}R\}\leqslant^{-1}\{P/\wedge^{-1}R\}$.

34.2.7 Inverse second-level single quasi-expert system towards the major premise

Suppose we have the classification tree of bronze, brass, or copper ware shown in Fig. 34.10.

This kind of mutually-inversistic machine learning is done as follows: if there is an upward route from vertex B to vertex A, then the conjunction of the first-order empirical or mathematical connection propositions denoted by each edge of the route is the minor premise, vertex B mutually inversely implying vertex A is the conclusion, the major premise learned

is the minor premise mutually inversely implying the conclusion. For example, there is an upward route from the leaf tripod to the root bronze_brass_copper_ware, the first-order single logical connection proposition learned is:

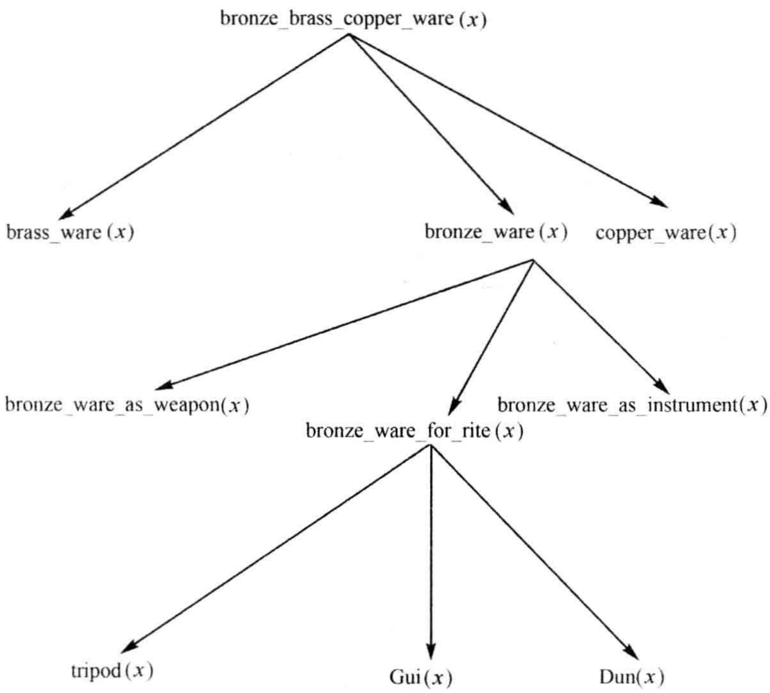


Fig. 34.10 classification tree of bronze, brass, or copper ware

$\{\text{tripod}(x) \leqslant^{-1} \text{bronze\_ware\_for\_rite}(x)\} \wedge \{\text{bronze\_ware\_for\_rite}(x) \leqslant^{-1} \text{bronze\_ware}(x)\} \wedge \{\text{bronze\_ware}(x) \leqslant^{-1} \text{bronze\_brass\_copper\_ware}(x)\} \leqslant^{-1} \{\text{tripod}(x) \leqslant^{-1} \text{bronze\_brass\_copper\_ware}(x)\}$.

34.2.8 Inverse second-level single quasi-expert system towards the minor premise

Suppose the classification tree is shown in Fig. 34.10. This kind of mutually-inversistic machine learning is done as follows: suppose there exists an upward route from vertex B to vertex A, then the conjunction of the first-order empirical or mathematical connection propositions denoted by each edge of the route is the minor premise learned; otherwise, nothing can be learned. For example, there exists an upward route from tripod to bronze-ware, therefore, we learn the minor premise:

$\{\text{tripod}(x) \leqslant^{-1} \text{bronze\_ware\_for\_rite}(x)\} \wedge \{\text{bronze\_ware\_for\_rite}(x) \leqslant^{-1} \text{bronze\_ware}(x)\}$.

Another example, there is no upward route from tripod to copper_ware, therefore, nothing can be learned.

34.2.9 Inverse first-level multiple decomposition system towards the major premise

34.2.9.1 Multiple connection operators association rule mining

For this kind of mutually-inversistic machine learning, see Chapter 35.

34.2.9.2 Inverse resolution

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.11, where comp means comparable.

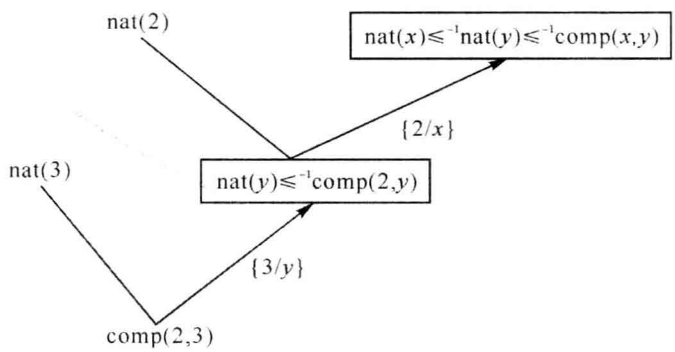


Fig 34.11 Inverse first-level multiple decomposition system towards the major premise

34.2.10 Inverse first-level multiple decomposition system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.12.

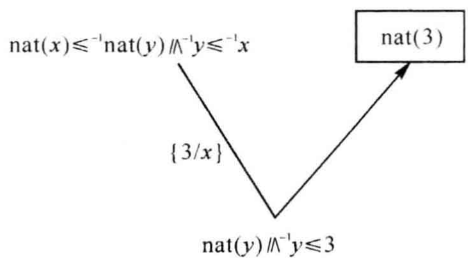


Fig. 34.12 Inverse first-level multiple decomposition system towards the minor premise

34.2.11 Inverse first-level multiple expert system towards the major premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.13.

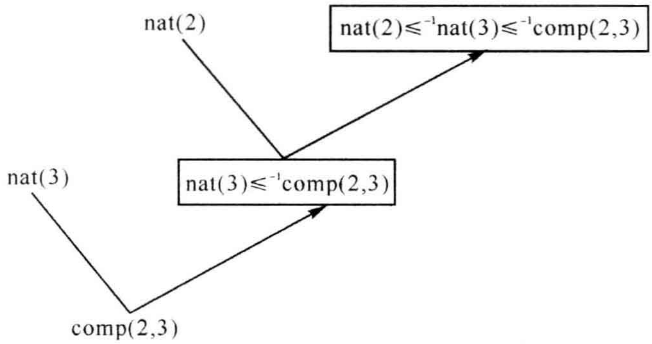


Fig. 34.13 Inverse first-level multiple expert system towards the major premise

34.2.12 Inverse first-level multiple expert system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.14.

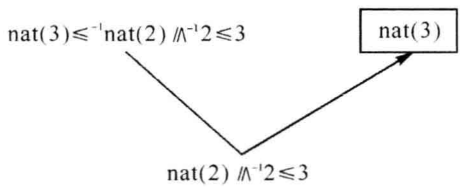


Fig. 34.14 Inverse first-level multiple expert system towards the minor premise

34.2.13 Inverse second-level multiple decomposition system towards the major premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.15.

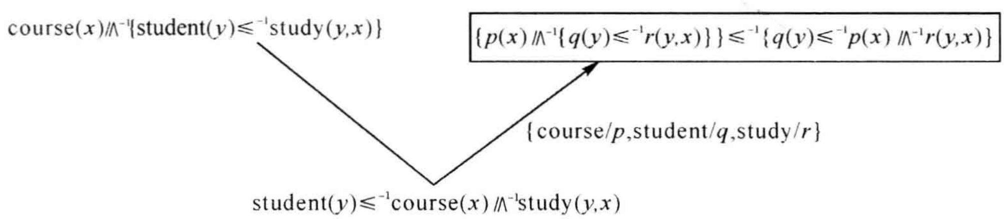


Fig. 34.15 Inverse second-level multiple decomposition system towards the major premise

34.2.14 Inverse second-level multiple decomposition system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.16.

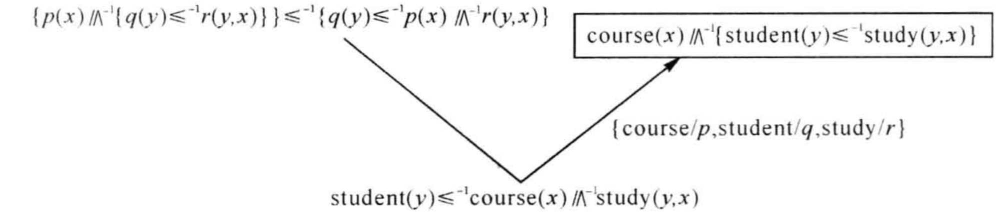


Fig. 34.16 Inverse second-level multiple decomposition system towards the minor premise

34.2.15 Inverse second-level multiple expert system towards the major premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.17.

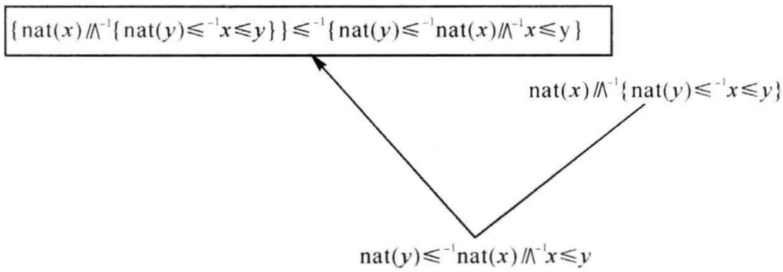


Fig. 34.17 Inverse second-level multiple expert system towards the major premise

34.2.16 Inverse second-level multiple expert system towards the minor premise

The inverse resolution of this kind of mutually-inversistic machine learning is shown in Fig. 34.18.

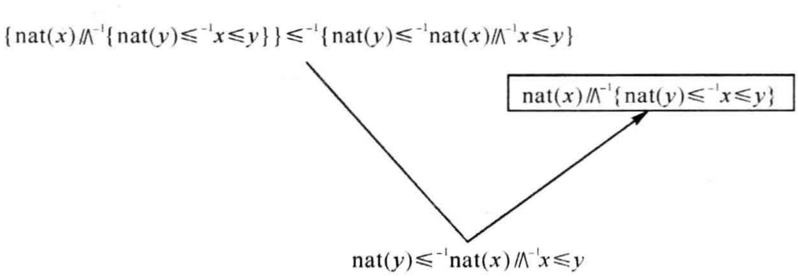


Fig. 34.18 Inverse second-level multiple expert system towards the minor premise

34.3 Summary on mutually-inversistic machine learning

Inductive learning, rough set learning, and genetic algorithm are the special cases of mutually-inversistic machine learning.

Various inverse second-level systems towards the major premise are from laws to learn laws of laws. They are first proposed in mutually-inversistic machine learning. Inverse second-level single quasi-expert system towards the major premise has been implemented as computer software.

Mutually-inversistic machine learning is useful. Inverse first-level multiple decomposition system towards the major premise is used in multiple connection operators association rule mining. Inverse second-level single quasi-decomposition system towards the minor premise is used in mutually-inversistic program refinement.

Chapter 35 Multiple connection operators association rule mining

35.1 Double connection operators association rule mining

35.1.1 Introduction to double connection operators association rule mining

Single dimensional association rule mining mines such rules as buys(X, "computer")→buys(X, "financial_management_software"), in which there is only one connection operator →. Multidimensional association rule mining mines such rules as age(X, "30...39")$\wedge$income(X, "42000...48000")→buys(X, "high_resolution_TV"), in which there is only one connection operator →. Single dimensional association rule mining and multidimensional association rule mining are called by a joint name single connection operator association rule mining.

In mutually-inversistic logic, such rule as p'(x)φ_1q'(y)φ_2r'(x, y) is called a double connection operators association rule, where φ_1 and φ_2 are connection operators $/\wedge^{-1}$ or $\leqslant^{-1}$, p'(x) and q'(y) are unary predicates, called property fact propositions, r'(x, y) is a binary predicate, called a nonproperty fact proposition. For example, course(Cno)$/\wedge^{-1}${student(Sno)$\leqslant^{-1}$ study(Sno, Cno)} is a double connection operators association rule, meaning that there exist a course Cno, such that for all students Sno, Sno study Cno. Another example is teacher(Tno)$\leqslant^{-1}$ course(Cno)$/\wedge^{-1}$teach(Tno, Cno), meaning that for all teachers Tno there exists a course Cno such that Tno teaches Cno.

35.1.2 Relational database of study and teaching

The E-R diagram of the relational database of study and teaching is shown in Fig. 35.1.

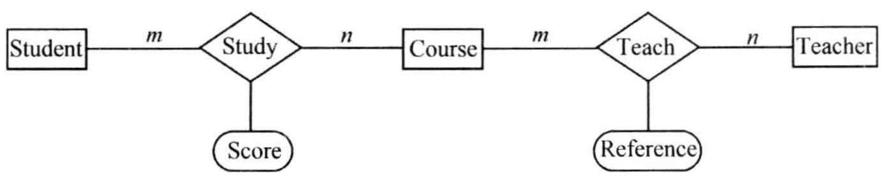


Fig. 35.1 The E-R diagram of the relational database of study and teaching

Transform the E-R diagram into relational model, we obtain the relational schemes:

Student(Sno, Sname, Age)

Course(Cno, Ctitle, Credit)

Study(Sno, Cno, Grade)

Teacher(Tno, Tname, Work_load)

Teach(Tno, Cno, Reference)

The primary keys of the relations are underlined. Some of the tables of the database are given in Tables 35.1 through 35.3.

Table 35.1
Student table

Sno	Sname	Age
S_1	Yong Li	20
S_2	Chen Liu	19
S_3	Min Wang	18
S_4	Yan Zhang	19
S_5	Ping Zhou	20

Table 35.2
Course table

Cno	Ctitle	Credit
C_1	Database	3
C_2	Mathematical analysis	4
C_3	General physics	2
C_4	Automatic control	3

Table 35.3
Study table

Sno	Cno	Grade
S_1	C_1	92
S_1	C_2	85
S_2	C_1	88
S_2	C_3	90
S_3	C_1	80
S_4	C_2	76
S_5	C_1	93
S_5	C_4	70

The student and course tables are obtained from the entities student and course of Fig. 35.1 respectively, and are called the entity tables. The study table is obtained from the binary relationship study of Fig. 35.1, and is called the binary relationship table. Take the primary key *Sno* of the entity table student and the primary key *Cno* of the entity table course as the property fact proposition, take the primary key (*Sno*, *Cno*) of the binary relationship table study as the nonproperty fact proposition, we obtain the double connection operators association rule student(*Sno*)$\varphi_1$course(*Cno*)$\varphi_2$study(*Sno*, *Cno*). This is to say, a relational database with a binary relationship naturally implies a double connection operators association rule.

35.1.3 Double connection operators association rule mining algorithm

We take student(*Sno*)$\varphi_1$course(*Cno*)$\varphi_2$study(*Sno*, *Cno*) as an example to investigate the double connection operators association rule mining algorithm. Let φ_1 and φ_2 be $/\wedge^{-1}$ and $\leqslant^{-1}$ respectively, we can obtain 6 double connection operators association rules (DCOAR):

DCOAR$_1$: student(*Sno*)$\leqslant^{-1}$course(*Cno*)$\leqslant^{-1}$study(*Sno*, *Cno*)

DCOAR$_2$: course(*Cno*)$/\wedge^{-1}$\{student(*Sno*)$\leqslant^{-1}$study(*Sno*, *Cno*)\}

DCOAR$_3$: student(*Sno*)$\leqslant^{-1}$course(*Cno*)$/\wedge^{-1}$study(*Sno*, *Cno*)

$DCOAR_4$: student(Sno)$/\wedge^{-1}${course(Cno)$\leqslant^{-1}$study(Sno, Cno)}

$DCOAR_5$: course(Cno)$\leqslant^{-1}$student(Sno)$/\wedge^{-1}$study(Sno, Cno)

$DCOAR_6$: student(Sno)$/\wedge^{-1}$course(Cno)$/\wedge^{-1}$study(Sno, Cno)

$\leqslant^{-1}$ means "for all", it require that all the instances satisfy the property fact proposition. This requirement is too high, we loosen its requirement by setting certainty factors c_{f1} and c_{f2} (they are percentages) for φ_1 and φ_2. If the percentage of the instances satisfying the property fact proposition is no less than the certainty factor, then $\leqslant^{-1}$ holds, and it does not mean "for all" but "many". If the percentage is less than the certainty factor, but at least one instance satisfies the property fact proposition, then $/\wedge^{-1}$ holds, and it means "some".

According to the binary relationship table study, we make the relational matrix shown in Fig. 35.2.

$$\begin{array}{cc} & \begin{array}{cccc} C_1 & C_2 & C_3 & C_4 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{array} & \begin{bmatrix} T & T & F & F \\ T & F & T & F \\ T & F & F & F \\ F & T & F & F \\ T & F & F & T \end{bmatrix} \end{array}$$

Fig. 35.2 Relational matrix for the binary relationship table

The relational matrix of Fig. 35.2 is made as follows: in Table 35.3 there is the tuple $<S_1, C_1>$, then at the intersection of S_1 and C_1 of the relational matrix a T is filled; in Table 35.3 there is no tuple $<S_1, C_3>$, then at the intersection of S_1 and C_3 of the relational matrix an F is filled.

Suppose the number of students is M, in this example 5; i.e., S_1 through S_5. Suppose the number of courses is N, in this example 4; i.e., C_1 through C_4. The algorithms of the double connection operators association rules mining are as follows:

$DCOAR_1$: If, in Fig. 35.2, there exists an $M*c_{f1}$ row and $N*c_{f2}$ column submatrix in which all elements are T, then DCOAR1 holds.

$DCOAR_2$: If, in Fig. 35.2, there exists at least one column in which there are no less than $M*c_{f1}$ Ts, then $DCOAR_2$ holds.

$DCOAR_3$: If, in Fig. 35.2, there are no less than $M*c_{f1}$ rows in which there is element T, then $DCOAR_3$ holds.

$DCOAR_4$: If, in Fig. 35.2, there is at least one row in which there are no less than $N*c_{f2}$ Ts, then $DCOAR_4$ holds.

$DCOAR_5$: If, in Fig. 35.2, there are no less than $N*c_{f2}$ columns in which there is element T, then $DCOAR_5$ holds.

$DCOAR_6$: If, in Fig. 35.2, there is at least one element T, then $DCOAR_6$ holds.

$DCOAR_1$ through $DCOAR_6$ form a complement lattice shown in Fig. 35.3.

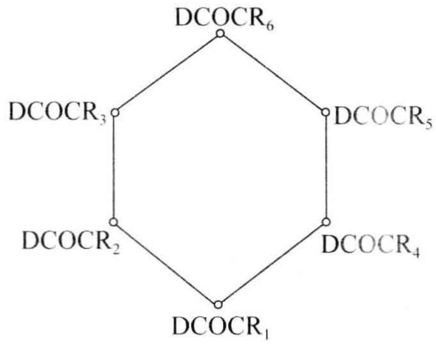


Fig. 35.3 The complement lattice formed by $DCOAR_1$ through $DCOAR_6$

In Fig. 35.3, the lower DCOAR implies the upper DCOAR; i.e., if $DCOAR_j$ is reachable from $DCOAR_i$ via an ascending path, and $DCOAR_i$ holds, then $DCOAR_j$ holds.

Since DCOAR1 through DCOAR6 satisfy Fig. 35.3, their algorithms can be merged into one, called $DCOAR_i$ determination algorithm, as is shown in Fig. 35.4.

Suppose c_{f1}=80%, c_{f2}=75%. In Fig. 35.2, in the column of C_1, there are $M*c_{f1}$=5*80%=4 T elements, therefore, $DCOAR_2$: course(Cno)$/\wedge^{-1}${student(Sno)$\leqslant^{-1}$study(Sno, Cno)} holds. From Fig. 35.3 we know that $DCOAR_3$ and $DCOAR_6$ also hold. In Fig. 35.2, in no less than$*c_{f2}$=4*75%=3 columns, there are T elements (for the column of C_1, there is S_1; for the column of C_2, there is S_1; for the column of C_3, there is S_2; for the column of C_4, there is S_5), therefore, $DCOAR_5$: course(Cno)$\leqslant^{-1}$student(Sno)$/\wedge^{-1}$study(Sno, Cno) holds.

35.1.4 Summary of double connection operators association rule mining

(1) Double connection operators association rule mining is different from single connection operator association rule mining. What single connection operator association rule mining mines is frequent itemsets, while what double connection operators association rule mining mines is the association among the primary keys of two entity tables and one binary relationship table.

(2) Fig. 35.2 is different from the data cube in data warehouse, the elements in the former are F and T, while the elements in the latter are data.

(3) There are three differences between double connection operators association rule mining and database quary: (i) the information to be queried by a database is anticipated, hence the SQL statements be written, while the information to be mined is unknown beforehand, and implied. (ii) Database query needs to write SQL statements, while mining is automated. (iii) The information queried is quantitative, while the information mined is qualitative, such as "many" and "some".

(4) If we view the dimension table of the star model of data warehouses as the entity table of relational databases, view the fact table as the binary relationship table, then this algorithm is also suitable for data warehouse mining.

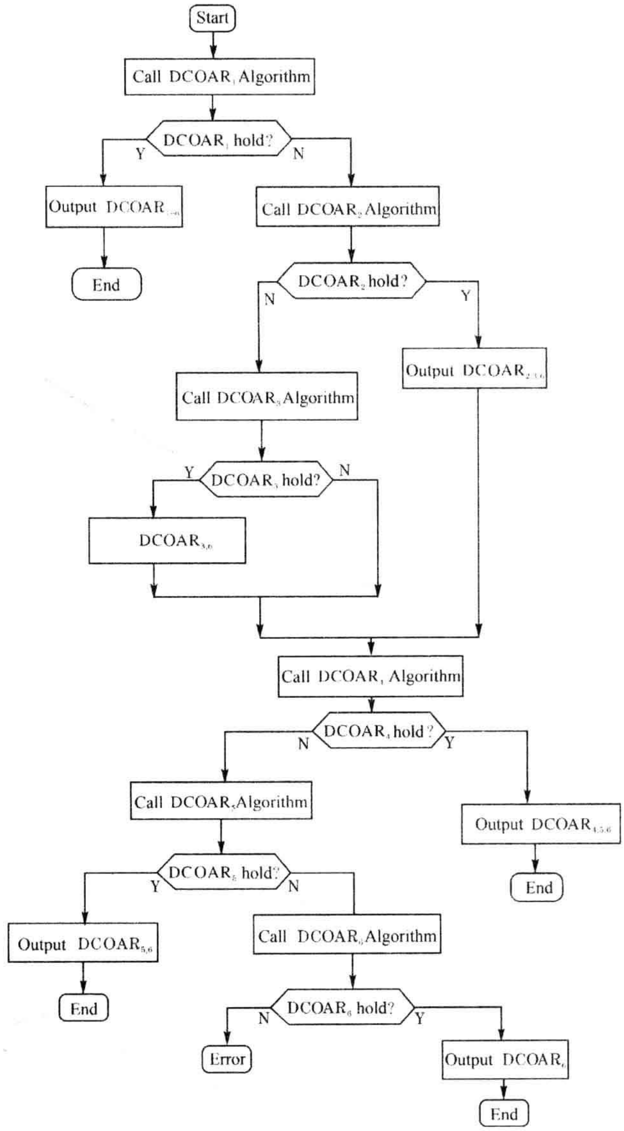


Fig. 35.4 DCOAR$_i$ determination algorithm

35.2 Triple connection operators association rule mining and degradation mining

35.2.1 Introduction to triple connection operators association rule mining

In mutually-inversistic logic, such rule as p’(x)φ_1q’(y)φ_2r’(z)φ_3s’(x, y, z) is called a

triple connection operators association rule, in which φ_1, φ_2, and φ_3 are connection operators $/\wedge^{-1}$ or $\leqslant^{-1}$, p'(x), q'(y), and r'(z) are unary predicates, called the property fact propositions, s'(x, y, z) is a ternary predicate, called the nonproperty fact proposition. For example, supplier(Sn)$/\wedge^{-1}$client(CLn)$/\wedge^{-1}$commodity(COn)$\leqslant^{-1}$purchase(Sn, CLn, COn) is a triple connection operators association rule, meaning that there exists supplier Sn, there exists client CLn, such that for all commodities COn, CLn purchases Sn's COn.

35.2.2 Relational database of purchase

The E-R diagram of the relational database of purchase is shown in Fig. 35.5.

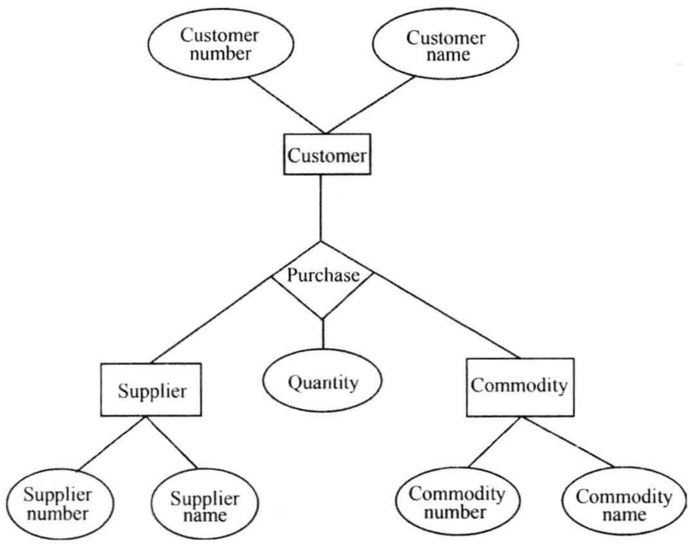


Fig. 35.5 E-R diagram of the relational database of purchase

Transform the E-R diagram into the relational model, we obtain the relational schemes:

Client(CLno, <u>CLn</u>)

Supplier(Sno, <u>Sn</u>)

Commodity(COno, <u>COn</u>)

Purchase(<u>CLn</u>, <u>Sn</u>, <u>COn</u>, quantity)

The primary keys of the relations are underlined. The tables of the relational database are shown in Tables 35.4 through 35.7.

Table 35. 4
Client table

CLno	CLn
Cli_1	Zhang
Cli_2	Li
Cli_3	Wang

Table 35.5
Supplier table

Sno	Sn
S_1	Hitachi
S_2	Samsung
S_3	Toshiba

Table 35.6
Commodity table

COno	COn
Com_1	TV
Com_2	frig
Com_3	washer

Table 35.7 Purchase table

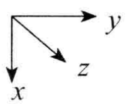
CLn	Sn	COn	quantity
Zhang	Hitachi	TV	1
Zhang	Hitachi	frig	1
Li	Hitachi	TV	1
Li	Hitachi	washer	1
Wang	Hitachi	frig	1
Wang	Samsung	washer	1

The client, supplier, and commodity tables are obtained from the entities of Fig. 35.5, and are called entity tables. The purchase table is obtained from the ternary relationship of Fig. 35.5, and is called a ternary relationship table. Take the primary key *CLn* of the entity table client, the primary key *Sn* of the entity table supplier, and the primary key *COn* of the entity table commodity as the property fact propositions, take the primary key (*CLn*, *Sn*, *COn*) of the ternary relationship table purchase as the nonproperty fact proposition, we obtain the triple connection operators association rule

client(CLn)φ1supplier(Sn)φ2commodity(COn)φ3purchase(CLn, Sn, COn). This is to say, a relational database with a ternary relationship naturally implies a triple connection operators association rule.

35.2.3 Triple connection operators association rule mining

According to Table 35.7, we make the cuboidal relational matrix shown in Fig. 35.6.



$$\begin{array}{c} \\ \text{Zhang} \\ \text{Li} \\ \text{Wang} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Hitachi} & \text{Samsung} & \text{Toshiba} \end{array} \\ \begin{bmatrix} T & F & F \\ T & F & F \\ F & F & F \end{bmatrix} \end{array} \text{TV}$$

$$\begin{array}{c} \\ \text{Zhang} \\ \text{Li} \\ \text{Wang} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Hitachi} & \text{Samsung} & \text{Toshiba} \end{array} \\ \begin{bmatrix} T & F & F \\ F & F & F \\ T & F & F \end{bmatrix} \end{array} \text{frig}$$

$$\begin{array}{c} \\ \text{Zhang} \\ \text{Li} \\ \text{Wang} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Hitachi} & \text{Samsung} & \text{Toshiba} \end{array} \\ \begin{bmatrix} F & F & F \\ T & F & F \\ F & T & F \end{bmatrix} \end{array} \text{Washer}$$

Fig. 35.6 Cuboidal relational matrix of Table 35.7

Fig. 35.6 is made as follows: there is tuple ⟨Zhang, Hitachi, TV⟩ in Table 35.7, then, at the intersection of Zhang, Hitachi, and TV of Fig. 35.6, a T is filled; there is no tuple ⟨Zhang, Hitachi, washer⟩, then, at the intersection of Zhang, Hitachi, and washer of Fig. 35.6, an F is filled.

Suppose the certainty factor is 66%. In Fig. 35.6, there is no 2*2*2 cuboidal submatrix with all its elements being T, therefore, client(CLn)$\leqslant^{-1}$supplier(Sn)$\leqslant^{-1}$commodity(COn)$\leqslant^{-1}$purchase(CLn, Sn, COn) does not hold. No supplier has a 2*2 subplane with all its elements being T, therefore, supplier(Sn)$/\wedge^{-1}${client(CLn)$\leqslant^{-1}$commodity(COn)$\leqslant^{-1}$purchase(CLn, Sn, COn) does not hold. The column determined by Zhang and Hitachi has two Ts, they are ⟨Zhang, Hitachi, TV⟩ and ⟨Zhang, Hitachi, frig⟩, therefore, client(CLn)$/\wedge^{-1}$supplier(Sn)$/\wedge^{-1}${commodity(COn)$\leqslant^{-1}$purchase(CLn, Sn, COn)} holds, meaning that there exists client CLn, there exists supplier Sn, such that for many commodity COn, CLn purchases Sn's COn.

If we view the purchase system as the data warehouse of star model, view Tables 35.4, 35,5, and 35,6 as dimension tables, view Table 35.7 as a fact table, then we can apply triple connection operators association rule mining to the data warehouse of star model.

35.2.4 Degradation mining

Tables 35.4 through 35.7 naturally implies triple connection operators association rule. If we apply double connection operators association rule mining to them, then it is called degradation mining.

35.2.4.1 Degradation mining in relational database and data warehouse of star model

Suppose we want to mine client(CLn)$\varphi_1$commodity(COn)$\varphi_2$purchase(CLn, COn). According to Table 35.7, we can make the relational matrix shown in Fig. 35.7.

$$\begin{array}{c} \\ \text{Zhang} \\ \text{Li} \\ \text{Wang} \end{array} \begin{array}{c} \begin{array}{ccc} \text{TV} & \text{frig} & \text{washer} \end{array} \\ \begin{bmatrix} \text{T} & \text{T} & \text{F} \\ \text{T} & \text{F} & \text{T} \\ \text{F} & \text{T} & \text{T} \end{bmatrix} \end{array}$$

Fig. 35.7 Degradation mining of Table 35.7

Suppose certainty factor is 66%. From Fig. 35.7, we can mine client(CLn)$/\wedge^{-1}${commodity(COn)$\leqslant^{-1}$purchase(CLn, COn)} and commodity(COn)$/\wedge^{-1}${client(CLn)$\leqslant^{-1}$purchase(CLn, COn).

35.2.4.2 Degradation mining in data cubes

Suppose we want to mine client(CLn)$\varphi_1$commodity(COn)$\varphi_2$purchase(CLn, COn) from the data cube of Fig. 35.6. Then we should mine the planes of Hitachi, Samsung, and Toshiba respectively. The plane of Hitachi is shown in Fig. 35.8.

$$\begin{array}{c c} & \begin{array}{ccc} \text{TV} & \text{frig} & \text{washer} \end{array} \\ \begin{array}{c} \text{Zhang} \\ \text{Li} \\ \text{Wang} \end{array} & \begin{bmatrix} \text{T} & \text{T} & \text{F} \\ \text{T} & \text{F} & \text{T} \\ \text{F} & \text{T} & \text{F} \end{bmatrix} \end{array}$$

Fig. 35.8 Degradation mining of Fig. 35.6

Suppose certainty factor is 66%. From Fig. 35.8, we can mine client(CLn)/$\wedge^{-1}$ {commodity(COn)$\leqslant^{-1}$purchase(CLn, COn)} and commodity(COn)/$\wedge^{-1}${client(CLn)$\leqslant^{-1}$ purchase(CLn, COn).

35.2.5 Summary of triple connection operators association rule mining and degradation mining

(1) The usual relationships in relational databases are binary relationship and ternary relationship. Therefore, double and triple connection operators association rule mining can mine most of the association rules implied in relational databases.

(2) A data warehouse has as less as 6 dimensions, as many as 15 dimensions. Therefore, only 6 to 15 connection operators association rule mining can mine the data warehouse precisely.

(3) The task of data mining is to mine the association rules that are finally understandable. Double connection operators association rule is understandable most easily. The more the connection operators an association rule has, the harder it is to understand it. Therefore, degradation mining is feasible.

(4) As to data warehouse, the degradation mining of star model and the degradation mining of data cube are different.

Chapter 36 Mutually-inversistic program refinement

36.1 Introduction to mutually-inversistic program refinement

There are many methods for program refinement, rule refinement method, VDM method, Z method, B method, etc.. Rule refinement method is a typical method. It has many refinement rules (inference rules). It is formalized. But since there are many refinement rules, it is hard for the system to choose from them automatically, therefore, it is not automated. Mutually-inversistic program refinement transforms the refinement rules into refinement axioms, takes second-level hypothetical inference and inverse second-level single quasi-decomposition system towards the minor premise as the only inference rules. And in a particular situation, a particular inference rule is used. The refinement axioms are searched top-down. In this way, program refinement can be done semi-automatically.

36.2 Introduction to rule refinement method

In order to read this chapter, the readers should be familiar with Chapters 1, 3, 4, and 5 of the literature (Morgan, 2002), which are introduced briefly in this section.

Program refinement is to refine the executable codes from the specifications. A specification is in the form of

w: [pre, post]

where pre denotes precondition, the initial state before a computer executes, post denotes the postcondition, the final state after a computer executes, w is the variable whose value may vary during a computer execution. For example, y:[$0\leqslant x\leqslant 9$, $y^2=x\wedge y\geqslant 0$] is a specification, which can be refined to the code y:= x (an assignment statement). This fact is denoted by;

y:[$0\leqslant x\leqslant 9$, $y^2=x\wedge y\geqslant 0$] $\sqsubseteq y:=\sqrt{x}$,

where $\sqsubseteq$ denotes "being refined as".

Usually, in the refinement process from the specification to the code, there are many intermediate states, which are the mixed program of specification and code.

If the precondition is true under all states, then it is denoted as

w:[true, post]

simplified as

w: [post].

The rule of assignment: if pre implies post[w\E], then

w, x: [pre, post] $\sqsubseteq$ w: =E,

where post denotes that all the occurrences of w in post are substituted by E.

Skip denotes that nothing is done.

Alternation is also called if statement. It realizes case analysis: according to the initial states, choose one command from several possible ones to execute. Alternation is constructed from a set of guarded commands. A guarded command is in the form of

G→prog

read "G then prog". Here, G is the guard (the condition to be satisfied), prog is the program to be executed after the condition is met.

An alternation is a collection of guarded commands, which are separated by the symbol $\square$ (read "otherwise"). The collection of the guarded commands is bracketed by the alternation brackets if and fi. The following is a general alternation:

if G_0→$prog_0$

$\square$ G_1→$prog_1$

⋮

$\square$ G_n→$prog_n$

fi

simplified as if ($\square i \cdot G_i \rightarrow prog_i$)fi.

The rule of alternation: if pre implies GG (GG is the abbreviation of $G_0 \vee G_1 \vee \ldots \vee G_n$), then

w:[pre, post] $\sqsubseteq$ if($\square i \cdot G_i \rightarrow w$:[$G_i \wedge$pre, post])fi

Now, let us consider the program of finding the maximum value. We use $\sqcup$ to denote "maximum". The specification is m:[m=a $\sqcup$ b]. The refinement process is

m:[m=a $\sqcup$ b]

$\sqsubseteq$ "according to the rule of alternation"

if a$\geqslant$b→m:[a$\geqslant$b, m=a$\sqcup$b] (1)

$\square$ b$\geqslant$a→m:[b$\geqslant$a, m=a$\sqcup$b] (2)

fi

(1) $\sqsubseteq$ "according to the rule of assignment" m:=a

(2) $\sqsubseteq$ "according to the rule of assignment" m:=b.

Iteration is sometimes called while loop. It realizes repetitive computation. The typical case of iteration is that when certain condition holds, execute a command repetitively. The most general form of iteration is constructed by the guarded commands, written as the following form:

do $G_0 \rightarrow prog_0$

□ $G_1 \rightarrow prog_1$

:

□ $G_n \rightarrow prog_n$

od

simplified as do(□$i \cdot G_i \rightarrow prog_i$)od.

It is known that 1, 2, 4, 8, ...are all the power of 2 (denoted as pt 2). We use $2|n$ to denote that 2 divides n evenly. In the following program, so long as the natural number n is the power of 2 initially, we get n=1 at last:

$$\text{do } 2|n \rightarrow n:=n\div 2 \text{ od} \tag{3}$$

Pt 2 holds before and after each time $n:=n\div 2$ is executed, therefore, pt 2 is called the invariant of the iteration. The guard $2|n$ will not hold at last. The iteration will terminate, because every time $n:=n\div 2$ is executed, the value of n decreases, and n cannot turn to the negative value. At this time we say that n is the variant of the iteration.

Program (3) is the refinement of the specification n:[pt n, n=1]:

n:[pt n, n=1]

$\sqsubseteq$ n:[pt n, pt $n \wedge \neg(2|n)$]

$\sqsubseteq$ do $2|n \rightarrow n:=n\div 2$ od.

36.3 Refinement axioms

In this section, the six refinement rules (inference rules) of rule refinement method are transformed into refinement axioms, which are logical axioms, and $\sqsubseteq$ is a logical connection operator. The six axioms, top-down, are:

The axiom of skip: if pre$\leqslant^{1}$post, then

w:[pre, post] $\sqsubseteq$ skip.

The axiom of assignment: if$(w=w_0)\wedge(x=x_0)\wedge$pre$\leqslant^{1}$post[$w$\E], then

w, x:[pre, post] $\sqsubseteq$ w:=E.

The axiom of simple specification: if E does not contain w, then

w:=E = w:[w=E].

The axiom of iteration: let invariant inv be any formula, variant V be any integer valued expression. If GG is the disjunction of all guards, then

w:[inv, inv $\wedge \neg$GG] $\sqsubseteq$ do(□$i \cdot G_i \leqslant^{-1} w$:[inv $\wedge G_i$, inv $\wedge(0\leqslant V < V_0)$])od.

Here, inv and G_i cannot contain initial variables, and expression V_0 is $V[w/w_0]$, denoting the initial value of variant V.

The axiom of alternation: if pre$\leqslant^{-1}$GG, then

w:[pre, post] $\sqsubseteq$ if(□$i \cdot G_i \leqslant^{-1} w$:[$G_i \wedge$pre, post])fi.

The axiom of sequential composition: for any formula mid

w:[pre, post] $\sqsubseteq$ w:[pre, mid]; w:[mid, post].

All of the six refinement axioms have "if" parts, these are the additional conditions that should be satisfied. Only if the additional conditions are satisfied, can the specifications be unified with the antecedents of the refinement axioms.

The assignment statements in the axiom of assignment should be set by the human being that run the system.

The axiom of simple specification is the bidirectional refinement.

If a specification is of the form w:[inv, inv$\wedge\neg$GG], then it unifies with the antecedent of the axiom of iteration automatically.

The disjunction GG of the guards of the axiom of alternation is first set by mutually-inversistic machine learning, if it does not work, then set by a human being that run the system.

As to the axiom of sequential composition, if a specification is of the form w:[true, P'$\wedge$Q'], where P' and Q' are fact propositions, then the specification can be refined to w:[true, P']; w:[P', P'$\wedge$Q'] or w:[true, Q']; w:[Q', P'$\wedge$Q'] automatically. The other forms of specification can be set by mutually-inversistic machine learning.

36.4 Mutually-inversistic program refinement system

Mutually-inversistic program refinement system contain the six refinement axioms mentioned in the previous section. A specification is the input. A code is the output. Second-level hypothetical inference with one antecedent and inverse second-level single quasi-decomposition system towards the minor premise are the two inference rules. A refinement axiom is the major premise, a specification or a mixed program (as an empirical or mathematical theorem) is the minor premise, one of the two inference rules is used, a mixed program or a code is inferred as the conclusion. The six refinement axioms are formalized, they are suitable to any specifications.

The system adopts forward reasoning. The reasoning process can be described by a refinement tree. The search strategy of the refinement tree is similar to that of Prolog: top-down, from left to right, bounded depth-first plus backtracking. If certain leaf of the refinement tree is a code, then the branch corresponding to the leaf is a success branch, otherwise, it is a failure branch. For the first five refinement axioms, second-level hypothetical inference with one antecedent is used. For the sixth refinement axiom, inverse second-level single quasi-decomposition system towards the minor premise is used. The numerous refinement axioms are unified top-down. The system is semiautomatic.

36.5 Examples

Example 36.1: Find the maximal value $a \sqcup b \sqcup c$ of the three numbers a, b, and c, where $\sqcup$ denotes "maximum".

Solution: The specification is m:[true, $m=a \sqcup b \sqcup c$]. The refinement tree is shown in Fig. 36.1.

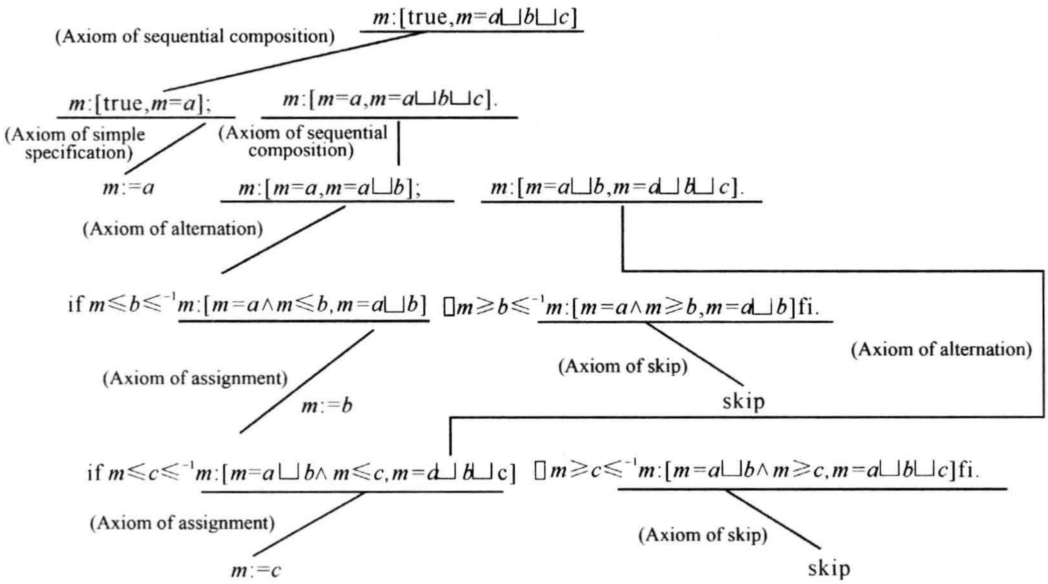


Fig. 36.1 Refinement tree for the Example 36.1

The search strategy of the refinement tree is similar to that of the SLD tree of Prolog. The only difference is that the search of the refinement tree is "bounded". But the understanding of the refinement tree is different from that of the SLD tree: for the refinement tree, the codes of the failure branches are not collected, the codes of the success branches are collected to obtain one solution; while for SLD tree, every success branch corresponds to a solution.

For Fig. 36.1, the codes of various success branches are collected, forming:

if $m \leqslant b \leqslant^{-1} m$:=b $\square$ $m \geqslant b \leqslant^{-1}$skip fi;

if $m \leqslant c \leqslant^{-1} m$:=c $\square$ $m \geqslant c \leqslant^{-1}$skip fi;

If we abbreviate if G $\leqslant^{-1}$prog $\square$ $\neg$G $\leqslant^{-1}$skip fi as

if G then prog fi

then the above codes can be written as

m:=a;

if $m \leqslant$b then m:=b fi;

if $m \leqslant$c then m:=c fi.

Example 36.2: Suppose natural number n is the power of 2. Every time n is divided by 2, at last n=1.

Solution: The specification is n:[pt n, pt $n \wedge \neg(2|n)$], where "pt" denotes the power of 2, "|" denotes dividing evenly. The refinement tree is shown in Fig. 36.2.

$$\underline{n\text{:[pt } n\text{,pt } n \wedge \neg(2|n)\text{]}}$$

(Axiom of iteration) |

$$\underline{\text{do } 2|n \leqslant^{-1} n\text{:[2 } n \wedge \text{ pt } n\text{,pt } n \wedge 0 \leqslant n \leqslant n_0\text{]}od}$$

(Axiom of assignment) |

$$n{:=}n \div 2$$

Fig. 36.2 Refinement tree for the Example 36.2

In Fig. 36.2, the check of the additional condition before the unification of the specification with the antecedent of the axiom of assignment is

$$n=n_0 \wedge 2|n \wedge \text{pt } n \leqslant^{-1} 0 \leqslant n < n_0 \wedge \text{pt } n \backslash n=n \div 2 \wedge 0 \leqslant n \div 2 < n_0 \wedge \text{pt}(n \div 2).$$

The expression satisfies the additional condition, therefore the unification can go on.

We collect the codes of Fig. 36.2, obtaining

do $2|n \leqslant^{-1} n{:=}n \div 2$ od.

We write the above codes with while statements, obtaining

```
while (2|n) do
    n:=n÷2.
```

36.6 Structure of solutions

A set of codes implementing the specification of a given problem is its solution. For a given problem, sometimes, its solution does not exist; if it exists, sometimes, it is not unique. For a given problem, the method for deciding whether the collected codes from the success branches of the refinement tree is a solution or not is to compute the number of the success branches: the number of success branch of the original problem is 1; every time the axiom of sequential composition is used, the number of success branches adds by 1; every time the axiom of alternation is used with the number of guards being n, the number of success branches add by $n - 1$.

Take Fig. 36.2 as an example. The number of success branch of the original problem is 1. The axiom of sequential composition is used twice, the axiom of alternation with the number of guards being 2 is used twice, therefore, its solution is composed of 5 success branches. In Fig. 36.2, there are just 5 success branches, they constitute a solution of the given problem.

36.7 Summary of mutually-inversistic program refinement

(1) Conventional program refinement can only be formalized, while mutually-inversistic program refinement can be semiautomatic.

(2) Mutually-inversistic program refinement has the refinement axioms of sequential composition, alternation, and iteration, can generate codes of structured programs.

(3) Mutually-inversistic program refinement and mutually-inversistic program verification are mutually inverse processes. In mutually-inversistic program refinement, when using the axiom of iteration to generate codes, invariants can be discovered, and invariant assertion needed by mutually-inversistic program verification can be formed at the same time, mutually-inversistic program refinement and mutually-inversistic program verification can be combined.

Part 14
Applications of mutually-inversistic mathematics

In Chapter 37, universal matrix is applied to OLAP of data warehouse, two-dimensional digital signal processing, and coordinate transformation. Chapter 38 introduces mutually-inversistic many-valued computer, including mutually-inversistic many-valued NAND gate and mutually-inversistic many-valued ANDORN gate. Mutually-inversistic many-valued NAND gate is based on mutually-inversistic abstract algebra. Mutually-inversistic many-valued ANDORN gate is based on mutually-inversistic abstract algebra and universal matrix. In Chapter 39, mutually-inversistic mathematical analysis is applied to modern control theory, experiment or observation data, and fuzzy controller.

Chapter 37 Applications of universal matrix

37.1 Applications of universal matrix to OLAP of data warehouse

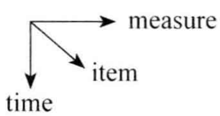
37.1.1 Introduction

In universal matrix, there are n-dimensional matrices, which can denote data cubes and supercubes in data warehouse. Various OLAP operations of data warehouse can be established on the operations of universal matrix.

37.1.2 Universal matrix representation of data cubes

37.1.2.1 Cuboidal matrix representation of data cubes

A data cube can be denoted by a cuboidal matrix. There are three ways of denoting a cuboidal matrix. Depth plane representation is the best way of denoting a cuboidal matrix, because each depth plane is just a page of a data cube. The depth plane representation of a data cube is shown in Fig. 37.1.



$$\begin{array}{c|cc|c} & \text{Sale} & \text{Cost} & \\ \hline \text{Jan.} & 450 & 350 & \\ \text{Feb.} & 380 & 280 & \text{Cap} \\ \text{Mar.} & 400 & 310 & \end{array}$$

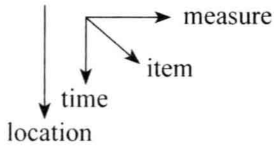
$$\begin{array}{c|cc|c} & \text{Sale} & \text{Cost} & \\ \hline \text{Jan.} & 550 & 450 & \\ \text{Feb.} & 460 & 360 & \text{Garmemt} \\ \text{Mar.} & 480 & 410 & \end{array}$$

$$\begin{array}{c|cc|c} & \text{Sale} & \text{Cost} & \\ \hline \text{Jan.} & 500 & 400 & \\ \text{Feb.} & 400 & 320 & \text{Jacket} \\ \text{Mar.} & 450 & 400 & \end{array}$$

Fig. 37.1 Depth plane representation of a data cube

37.1.2.2 4-deimensional matrix representation of data supercubes

The 4-dimensional matrix representation of a data supercube is shown in Fig. 37.2.



$$\text{New York}\left\{\begin{array}{l} \begin{array}{c|cc} & \text{Sale} & \text{Cost} \\ \hline \text{Jan.} & 220 & 170 \\ \text{Feb.} & 190 & 140 \end{array}\ \text{Cap} \\ \qquad\quad \begin{array}{c|cc} & \text{Sale} & \text{Cost} \\ \hline \text{Jan.} & 270 & 220 \\ \text{Feb.} & 230 & 180 \end{array}\ \text{Garment} \end{array}\right\}$$

$$\text{Boston}\left\{\begin{array}{l} \begin{array}{c|cc} & \text{Sale} & \text{Cost} \\ \hline \text{Jan.} & 200 & 160 \\ \text{Feb.} & 180 & 130 \end{array}\ \text{Cap} \\ \qquad\quad \begin{array}{c|cc} & \text{Sale} & \text{Cost} \\ \hline \text{Jan.} & 240 & 200 \\ \text{Feb.} & 220 & 170 \end{array}\ \text{Garment} \end{array}\right\}$$

Fig. 37.2 4-dimensional matrix representation of a data supercube

37.1.3 OLAP operations based on the operations of universal matrix

37.1.3.1 Slice

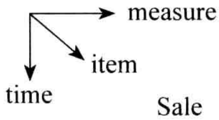
Slice operation can be based on selecting a plane from a cuboidal matrix. For example, selecting a depth plane from Fig. 37.1 is shown in Fig. 37.3.

$$\begin{array}{c} \\ \text{Jan.} \\ \text{Feb.} \\ \text{Mar.} \end{array} \begin{array}{c} \begin{array}{cc} \text{Sale} & \text{Cost} \end{array} \\ \begin{bmatrix} 550 & 450 \\ 460 & 360 \\ 480 & 410 \end{bmatrix} \end{array} \ \text{Garmemt}$$

Fig. 37.3 Slice operation

37.1.3.2 Dice

Dice operation can be based on selecting a submatrix from a cuboidal matrix. For example, selecting a submatrix from Fig. 37.1 is shown in Fig. 37.4.



$$\begin{array}{rc} & \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{array}{r}\text{Jan.}\\ \text{Feb.}\end{array} & \begin{bmatrix}450 & 350\\ 380 & 280\end{bmatrix}\end{array}\ \text{Cap}$$

$$\begin{array}{rc} & \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{array}{r}\text{Jan.}\\ \text{Feb.}\end{array} & \begin{bmatrix}550 & 450\\ 160 & 360\end{bmatrix}\end{array}\ \text{Garment}$$

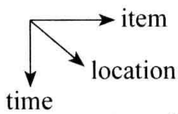
Fig. 37.4 Dice operation

37.1.3.3 Drill-up

There are two kind of drill-up operations. One is lifting up a concept level to a dimension. The other is dimension reduction.

37.1.3.3.1 Lifting up a concept level to a dimension

Suppose we have a data cube shown as a cuboidal matrix of Fig. 37.5.



$$\begin{array}{rc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{hone}\end{array} \\ \begin{array}{r}Q_1\\ Q_2\\ Q_3\\ Q_4\end{array} & \begin{bmatrix}180 & 200 & 220 & 240\\ 260 & 280 & 300 & 320\\ 340 & 360 & 380 & 400\\ 420 & 440 & 460 & 480\end{bmatrix}\end{array}\ \text{Canada}$$

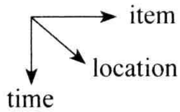
$$\begin{array}{rc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{phone}\end{array} \\ \begin{array}{r}Q_1\\ Q_2\\ Q_3\\ Q_4\end{array} & \begin{bmatrix}820 & 840 & 860 & 880\\ 900 & 920 & 940 & 960\\ 980 & 1000 & 1020 & 1040\\ 1060 & 1080 & 1100 & 1120\end{bmatrix}\end{array}\ \text{USA}$$

$$\begin{array}{rc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{phone}\end{array} \\ \begin{array}{r}Q_1\\ Q_2\\ Q_3\\ Q_4\end{array} & \begin{bmatrix}330 & 340 & 350 & 360\\ 370 & 380 & 390 & 400\\ 410 & 420 & 430 & 440\\ 450 & 460 & 470 & 480\end{bmatrix}\end{array}\ \text{New York}$$

$$\begin{array}{rc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{phone}\end{array} \\ \begin{array}{r}Q_1\\ Q_2\\ Q_3\\ Q_4\end{array} & \begin{bmatrix}490 & 500 & 510 & 520\\ 530 & 540 & 550 & 560\\ 510 & 580 & 590 & 600\\ 610 & 620 & 630 & 640\end{bmatrix}\end{array}\ \text{Chicago}$$

Fig. 37.5 A data cube

The drill-up operation by lifting up from the city level to the country level to the location dimension of Fig. 37.5 is shown in Fig. 37.6.



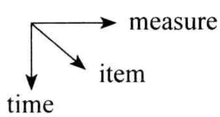
$$\begin{array}{cc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{phone}\end{array} \\ \begin{array}{c}Q_1\\Q_2\\Q_3\\Q_4\end{array} & \begin{bmatrix}180 & 200 & 220 & 240\\ 260 & 280 & 300 & 320\\ 340 & 360 & 380 & 400\\ 420 & 440 & 460 & 480\end{bmatrix}\end{array}\ \text{Canada}$$

$$\begin{array}{cc} & \begin{array}{cccc}\text{TV} & \text{frig} & \text{washer} & \text{phone}\end{array} \\ \begin{array}{c}Q_1\\Q_2\\Q_3\\Q_4\end{array} & \begin{bmatrix}820 & 840 & 860 & 880\\ 900 & 920 & 940 & 960\\ 980 & 1000 & 1020 & 1040\\ 1060 & 1080 & 1100 & 1120\end{bmatrix}\end{array}\ \text{USA}$$

Fig. 37.6 Drill-up operation from the city level to the country level to the location dimension of Fig. 37.5

37.1.3.3.2 Dimension reduction

The drill-up operation by reducing the location dimension of Fig. 37.2 is shown in Fig. 37.7.



$$\begin{array}{cc} & \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{array}{c}\text{Jan.}\\ \text{Feb.}\end{array} & \begin{bmatrix}420 & 330\\ 370 & 270\end{bmatrix}\end{array}\ \text{Cap}$$

$$\begin{array}{cc} & \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{array}{c}\text{Jan.}\\ \text{Feb.}\end{array} & \begin{bmatrix}510 & 420\\ 450 & 350\end{bmatrix}\end{array}\ \text{Garment}$$

Fig. 37.7 Drill-up operation by reducing location dimension of Fig. 37.2

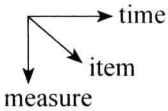
37.1.3.4 Rotation

There are three kind of rotation operations. The first kind is rotating the axes of each

two-dimensional slice. The second kind is rotating the axis of a three-dimensional data cube. The third kind is separating a three-dimensional data cube into a series of two-dimensional planes.

37.1.3.4.1 Rotating the axes of each two-dimensional slice

Rotating the axes of each two-dimensional slice can be based on the plane transpose of cuboidal matrix. For example, the depth plane transpose of the cuboidal matrix of Fig. 37.1 is shown in Fig. 37.8.

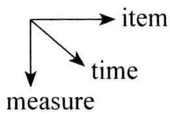


$$\begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Jan.} & \text{Feb.} & \text{Mar.}\end{array} \\ \begin{bmatrix} 450 & 380 & 400 \\ 350 & 270 & 310 \end{bmatrix} \end{array} \text{Cap} \quad \begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Jan.} & \text{Feb.} & \text{Mar.}\end{array} \\ \begin{bmatrix} 550 & 460 & 480 \\ 450 & 360 & 410 \end{bmatrix} \end{array} \text{Garment} \quad \begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Jan.} & \text{Feb.} & \text{Mar.}\end{array} \\ \begin{bmatrix} 500 & 400 & 450 \\ 400 & 320 & 400 \end{bmatrix} \end{array} \text{Jacket}$$

Fig. 37.8 Depth plane transpose of Fig. 37.1

37.1.3.4.2 Rotating the axis of a three-dimensional data cube

Rotating the axis of a three-dimensional data cube can be based on the right-shift or left-shift transpose of a cuboidal matrix. For example, the right-shift transpose of the cuboidal matrix of Fig. 37.1 is shown in Fig. 37.9.



$$\begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Cap} & \text{Garment} & \text{Jacket}\end{array} \\ \begin{bmatrix} 450 & 550 & 500 \\ 352 & 450 & 400 \end{bmatrix} \end{array} \text{Jan.}$$

$$\begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Cap} & \text{Garment} & \text{Jacket}\end{array} \\ \begin{bmatrix} 380 & 460 & 400 \\ 280 & 360 & 320 \end{bmatrix} \end{array} \text{Feb.}$$

$$\begin{array}{c} \\ \text{Sale} \\ \text{Cost} \end{array}\begin{array}{c} \begin{array}{ccc}\text{Cap} & \text{Garment} & \text{Jacket}\end{array} \\ \begin{bmatrix} 400 & 480 & 450 \\ 310 & 410 & 400 \end{bmatrix} \end{array} \text{Mar.}$$

Fig. 37.9 Right-shift transpose of Fig. 37.1

37.1.3.4.3 Seperating a three-dimensional data cube into a series of two-dimensional planes

Seperating each depth plane of Fig. 37.1 is shown in Fig. 37.10.

$$\begin{array}{c} \\ \text{Jan.} \\ \text{Feb.} \\ \text{Mar.} \end{array}\begin{array}{c} \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{bmatrix} 450 & 250 \\ 380 & 280 \\ 400 & 310 \end{bmatrix} \\ \text{Cap} \end{array} \quad \begin{array}{c} \\ \text{Jan.} \\ \text{Feb.} \\ \text{Mar.} \end{array}\begin{array}{c} \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{bmatrix} 550 & 450 \\ 460 & 360 \\ 480 & 410 \end{bmatrix} \\ \text{Garment} \end{array} \quad \begin{array}{c} \\ \text{Jan.} \\ \text{Feb.} \\ \text{Mar.} \end{array}\begin{array}{c} \begin{array}{cc}\text{Sale} & \text{Cost}\end{array} \\ \begin{bmatrix} 500 & 400 \\ 400 & 320 \\ 450 & 400 \end{bmatrix} \\ \text{Jacket} \end{array}$$

Fig. 37.10 Seperating each depth plane of Fig. 37.1

37.2 Application of universal matrix to two-dimensional digital signal processing

Universal matrix can be used in two-dimensional digital signal processing, shown in Fig. 37.11.

$$\begin{bmatrix} X(0,0) & X(0,1) \\ X(1,0) & X(1,1) \end{bmatrix} = \begin{array}{l} \begin{bmatrix} H(0,0,0) & H(0,1,0) \\ H(1,0,0) & H(1,1,0) \end{bmatrix} \\ \qquad \begin{bmatrix} H(0,0,1) & H(0,1,1) \\ H(1,0,1) & H(1,1,1) \end{bmatrix} \end{array} \quad \begin{bmatrix} x(0,0) & x(0,1) \\ x(1,0) & x(1,1) \end{bmatrix}$$

Fig. 37.11 Universal matrix used in two-dimensional digital signal processing

In Fig. 37.11, x matrix is the two-dimensional time domain signal matrix, X matrix is the two-dimensional frequency domain signal matrix, H matrix is the three-dimensional fast Fourier transformation matrix.

37.3 Application of universal matrix to coordinate transformation

Suppose $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$, $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$, and $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ are three coordinate systems in a 4-dimensional space. The interpretation of $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ is as follows: x_1Ox_2 is the displacement plane, $\dot{x}_1O\dot{x}_2$ is the velocity plane, $x_1O\dot{x}_1$ is the x_1 phase plane, $x_2O\dot{x}_2$ is the x_2 phase plane. Similar interpretations can be given for $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$ and $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$. The transformation between $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$ and $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$ is given in (37.1):

$$\begin{cases} z_1=a_{111}y_1+a_{112}y_2 \\ \dot{z}_1=a_{211}y_1+a_{212}y_2 \end{cases} \quad \begin{cases} z_2=a_{121}\dot{y}_1+a_{122}\dot{y}_2 \\ \dot{z}_2=a_{221}\dot{y}_1+a_{222}\dot{y}_2 \end{cases} \tag{37.1}$$

The transformation between $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ and $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$ is given in (37.2):

$$\begin{cases} y_1=b_{111}x_1+b_{112}x_2 \\ y_2=b_{121}x_1+b_{122}x_2 \end{cases} \qquad \begin{cases} \dot{y}_1=b_{211}\dot{x}_1+b_{212}\dot{x}_2 \\ \dot{y}_2=b_{221}\dot{x}_1+b_{222}\dot{x}_2 \end{cases} \tag{37.2}$$

Substituting (37.2) into (37.1), we obtain the transformation between $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ and $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$ given in (37.3):

$$\begin{cases} z_1=(a_{111}b_{111}+a_{112}b_{121})x_1+(a_{111}b_{112}+a_{112}b_{122})x_2 \\ \dot{z}_1=(a_{211}b_{111}+a_{212}b_{121})x_1+(a_{211}b_{112}+a_{212}b_{122})x_2 \end{cases}$$
$$\begin{cases} z_2=(a_{121}b_{211}+a_{122}b_{221})\dot{x}_1+(a_{121}b_{212}+a_{122}b_{222})\dot{x}_2 \\ \dot{z}_2=(a_{221}b_{211}+a_{222}b_{221})\dot{x}_1+(a_{221}b_{212}+a_{222}b_{222})\dot{x}_2 \end{cases} \tag{37.3}$$

Let

$$\begin{cases} z_1=c_{111}x_1+c_{112}x_2 \\ \dot{z}_1=c_{211}x_1+c_{212}x_2 \end{cases} \qquad \begin{cases} z_2=c_{1211}\dot{x}_1+c_{1222}\dot{x}_2 \\ \dot{z}_2=c_{2211}\dot{x}_1+c_{2222}\dot{x}_2 \end{cases} \tag{37.4}$$

Comparing (37.3) with (37.4), we obtain (37.5):

$$\begin{cases} c_{111}=a_{111}b_{111}+a_{112}b_{121} & c_{112}=a_{111}b_{112}+a_{112}b_{122} \\ c_{211}=a_{211}b_{111}+a_{212}b_{121} & c_{212}=a_{211}b_{112}+a_{212}b_{122} \end{cases}$$
$$\begin{cases} c_{121}=a_{121}b_{211}+a_{122}b_{221} & c_{122}=a_{121}b_{212}+a_{122}b_{222} \\ c_{221}=a_{221}b_{211}+a_{222}b_{221} & c_{222}=a_{221}b_{212}+a_{222}b_{222} \end{cases} \tag{37.5}$$

Formula (37.5) can be written in a general form (37.6):

$$c_{ijl}=\sum_{k=1}^{2} a_{ijk}b_{jkl}(i, j, l=1, 2) \tag{37.6}$$

We use the cuboidal matrices

$$A=\begin{bmatrix} a_{111} & a_{112} \\ a_{211} & a_{212} \end{bmatrix}\begin{bmatrix} a_{111} & a_{112} \\ a_{211} & a_{212} \end{bmatrix} \qquad B=\begin{matrix} \begin{bmatrix} b_{111} & b_{121} \\ b_{112} & b_{122} \end{bmatrix} \\ \begin{bmatrix} b_{211} & b_{221} \\ b_{212} & b_{222} \end{bmatrix} \end{matrix}$$

to denote the transformation between $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$ and $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$, the transformation between $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ and $\{O\text{-}y_1, y_2, \dot{y}_1, \dot{y}_2\}$ respectively, then the cuboidal matrix

$$C=\begin{bmatrix} c_{111} & c_{112} \\ c_{211} & c_{212} \end{bmatrix}\begin{bmatrix} c_{121} & c_{122} \\ c_{221} & c_{222} \end{bmatrix}$$

the transformation between $\{O\text{-}x_1, x_2, \dot{x}_1, \dot{x}_2\}$ and $\{O\text{-}z_1, z_2, \dot{z}_1, \dot{z}_2\}$, is just $A*_{rc}B$ determined by (37.6).

Chapter 38 Mutually-inversistic many-valued computer

In this chapter, we will study mutually-inversistic many-valued NAND gate and mutually-inversistic many-valued ANDORN gate, which have two schemes respectively: total ordering scheme and partial ordering scheme. Mutually-inversistic many-valued NAND gate is based on mutually-inversistic abstract algebra. Mutually-inversistic many-valued ANDORN gate is based on mutually-inversistic abstract algebra and universal matrix.

38.1 Mutually-inversistic many-valued NAND gates

38.1.1 Min-compl gate

Min-compl gate is a total ordering based n-valued NAND gate. N can be any positive integer greater than 2. We adopt n=3 (the most easily implementable many-valued NAND gate); i.e., the three values 0, 1, and 2. The total ordering is $\leqslant$. The carrying set is $\{0, 1, 2\}$. The operations min and max make a complement lattice to $\leqslant$. The complement operation ' is: 0'=2, 1'=1, 2'=0. We take min as the AND operation, take '(denoted by compl) as the NOT operation, thus, min-compl constitutes the total ordering based NAND gate. The mutually inverse diagram of the min-compl gate is composed of the vertices marked with " △ " of Fig. 38.1.

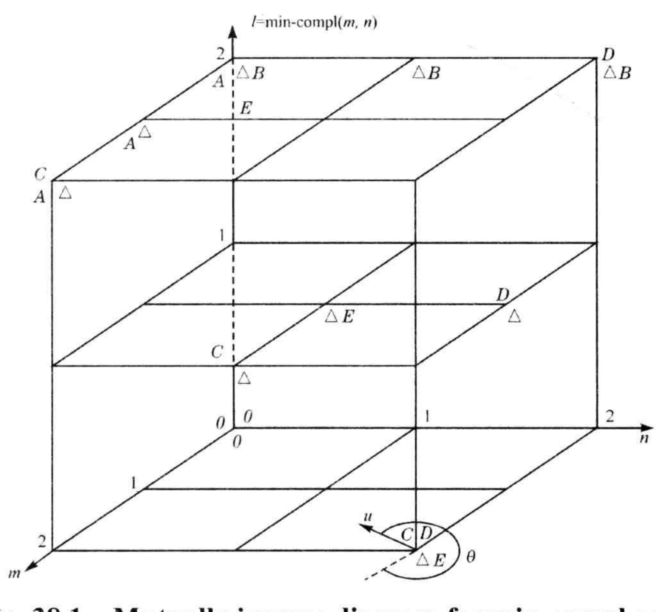


Fig. 38.1 Mutually inverse diagram for min-compl gate

In Fig. 38.1, every transaxis straight line denotes a double-sided discrete partial derivative, reveals certain algebraic property. The transaxis straight line marked with A denotes the partial derivative $\frac{\partial l}{\partial m}\Big|_{n=0}=0$, reveals the algebraic property min-compl(m, 0)=0'; i.e., n=0 is the right complement zero element. The transaxis straight line marked with B denotes the partial derivative $\frac{\partial l}{\partial n}\Big|_{m=0}=0$, reveals the algebraic property that m=0 is the left complement zero element. The transaxis straight line marked with C denotes the partial derivative $\frac{\partial l}{\partial n}\Big|_{m=2}=-1$, reveals the algebraic property min-compl(2, n)=n'; i.e., m=2 is the left complement identity element. The transaxis straight line marked with D denotes the partial derivative $\frac{\partial l}{\partial m}\Big|_{n=2}=-1$, reveals the algebraic property that n=2 is the right complement identity element. The transaxis straight line marked with E denotes the directional derivative $D_u$min-compl(m, n)$=\frac{\partial l}{\partial m}\cos\theta+\frac{\partial l}{\partial n}\sin\theta=\sqrt{2}$, reveals the complement idempotent law min-compl(n, n)=n'. If the transaxis straight lines A, C, and E are regarded as the vectors $\vec{A}$, $\vec{C}$, and $\vec{E}$, then $\vec{E}=\vec{A}+\vec{C}$. Likewise, $\vec{E}=\vec{B}+\vec{D}$.

38.1.2 GCD-compl gate

GCD-compl gate is a partial ordering based n-valued NAND gate. N adopts 2^i (i is any positive integer). We adopt $n=2^2=4$; i.e., the four values 1, 2, 3, and 6, which are the factors of 6. The partial ordering is dividing evenly relation. The carrying set is {1, 2, 3, 6}. The operations GCD (greatest common divisor), LCM (least common multiplier), and compl

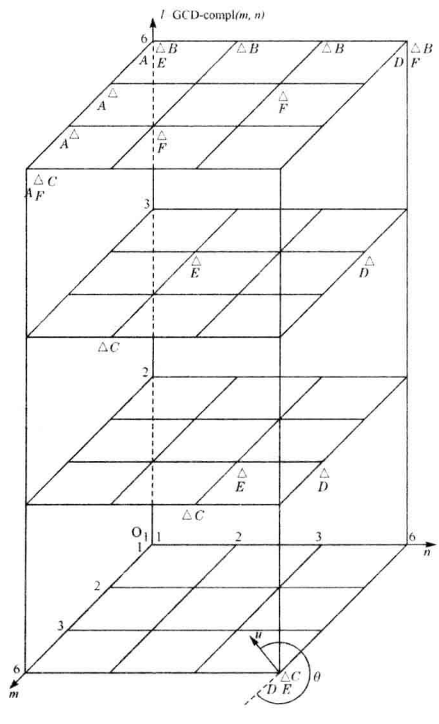


Fig. 38.2 Mutually inverse diagram for GCD-compl gate

(complement) make a Boolean algebra to the dividing evenly relation. The complement operation is: 1'=6, 2'=3, 3'=2, 6'=1. We take GCD as the AND operation, take compl as the NOT operation, thus, GCD-compl constitutes a partial ordering based NAND gate, in which 2 and 3 are not comparable, because 2 cannot divides 3 evenly, and 3 cannot divides 2 evenly. The mutually inverse diagram of GCD-compl gate is composed of the vertices marked "△" of Fig. 38.2.

In Fig. 38.2, the derivative denoted by, the algebraic property revealed by the transaxis straight lines A, B, C, D, and E are the same as those in Fig. 38.1. But Fig. 38.2 has one more transaxis straight line: line F, which denotes the partial derivative $\frac{\partial m}{\partial n}\Big|_{1=6}=-1$, reveals the algebraic property GCD-compl(n, n')=6 (the total upper bound); i.e., the law of total upper bound. In addition to $\vec{E}=\vec{A}+\vec{C}$ and $\vec{E}=\vec{B}+\vec{D}$, Fig. 38.2 has one more sum of vectors: $\vec{F}=\vec{A}+\vec{B}$.

38.1.3 Comparison between min-compl gate and GCD-compl gate

The advantages of min-compl gate over GCD-compl gate are: min-comple gate is based on total ordering, its values adopted are natural; it can be three-valued logic, the most easily implementable many-valued logic. The advantage of GCD-compl gate over min-compl gate is: GCD-compl embraces more information: one more partial derivative, one more sum of vectors.

38.2 Mutually-inversistic many-valued ANDORN gates

38.2.1 Introduction

Two-valued ANDORN gate is $m=n_1 \cdot n_2+n_3 \cdot n_4$, its symbol is shown in Fig. 38.3, its truth table is shown in Table 38.1.

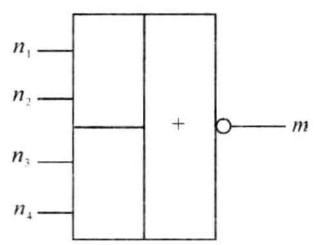


Fig. 38.3 Symbol for two-valued ANDORN gate

Table 38.1

Truth table for two-valued ANDORN gate

n_1	n_2	n_3	n_4	m
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Two-valued ANDORN gate can be denoted by the 4-dimensional universal matrix shown in Fig. 38.4.

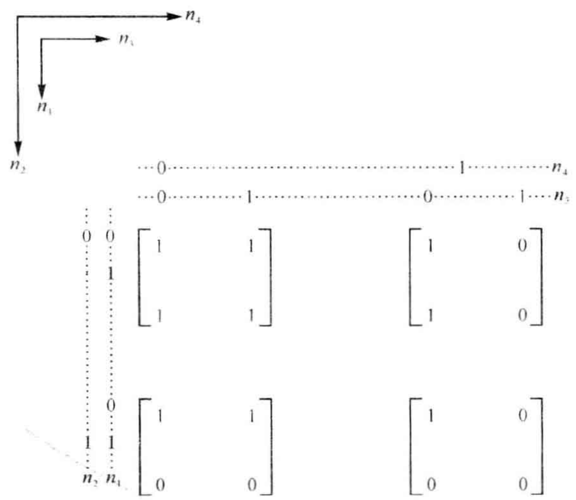


Fig. 38.4 4-dimensional universal matrix representation for two-valued ANDORN gate

The bottom left corner value 0 in the bottom left corner plane matrix of Fig. 38.4 is obtained as follows: n_1=1 and n_2=1, therefore $n_1 \wedge n_2$=1; n_3=0 and n_4=0, therefore, $n_3 \wedge n_4$=0; hence $(n_1 \wedge n_2) \vee (n_3 \wedge n_4)$=1∨0=1; hence $\neg((n_1 \wedge n_2) \vee (n_3 \wedge n_4))$= $\neg$(1)=0. The other values in the matrix can be similarly obtained. The universal matrix representation of two-valued ANDORN gate can be generalized to n-valued ANDORN gate.

38.2.2 Min-max-compl gate

38.2.2.1 Universal matrix representation of min-max-compl gate

Min-max-compl gate is a total ordering based n-valued ANDORN gate. N can be any positive integer greater than 2. We adopt n=3 (the most easily implementable many-valued ANDORN gate); i.e., the three values 0, 1, and 2. The total ordering is $\leqslant$. The carrying set is {0, 1, 2}. The operations min and max make a complement lattice to $\leqslant$. The complement operation ' is: 0'=2, 1'=1, 2'=0. We take min as the AND operation, take max as the OR operation, take '(denoted by compl) as the NOT operation, thus, min-max-compl constitutes the total ordering based ANDORN gate. Its symbol is shown in Fig. 38.5, its universal matrix representation is shown in Fig. 38.6.

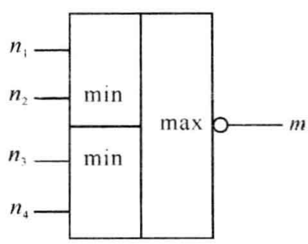


Fig. 38.5 Symbol for min-max-compl gate

$$
\begin{array}{c|ccc}
 & n_4 = 0 & 1 & 2 \\
\hline
n_2 = 0 &
\begin{array}{cc} & \begin{array}{ccc} n_3=0 & 1 & 2 \end{array} \\ \begin{array}{c} n_1=0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 0 \end{bmatrix} \end{array} \\
1 &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 2 & 2 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \end{array} \\
2 &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 2 & 2 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} &
\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}
\end{array}
$$

Fig. 38.6 Universal matrix representation of min-max-compl gate

Fig. 38.6 has four kinds of operations, described below.

38.2.2.2 Total lower bound type operations

Of the three values 0, 1, and 2, 0 is the total lower bound. The general form of min-max-compl operations can be denoted as

$m=f(n_1, n_2, n_3, n_4)$.

In Fig. 38.6, there are two operations with the results being the total lower bound 0:

$m=\mathrm{f}(2, 2, n_3, n_4)=0$

and

$m=\mathrm{f}(n_1, n_2, 2, 2)=0$.

$M=\mathrm{f}(2, 2, n_3, n_4)=0$ is the bottom row of Fig. 38.6. $M=\mathrm{f}(n_1, n_2, 2, 2)=0$ is the rightmost column of Fig. 38.6. $M=\mathrm{f}(2, 2, n_3, n_4)=0$ has two partial derivatives:

$\frac{\partial \mathrm{f}}{\partial \mathrm{n}3}\Big|_{\mathrm{n}1=2,\ \mathrm{n}2=2}=0$ (denoting that when $n_1=2$ and $n_2=2$, f does not vary with the variation of n_3, but remains 0 invariably), and $\frac{\partial \mathrm{f}}{\partial \mathrm{n}4}\Big|_{\mathrm{n}1=2,\ \mathrm{n}2=2}=0$.

$M=\mathrm{f}(n_1, n_2, 2, 2)=0$ has also two partial derivatives:

$\frac{\partial \mathrm{f}}{\partial \mathrm{n}1}\Big|_{\mathrm{n}3=2,\ \mathrm{n}4=2}=0$ and $\frac{\partial \mathrm{f}}{\partial \mathrm{n}2}\Big|_{\mathrm{n}3=2,\ \mathrm{n}4=2}=0$.

38.2.2.3 Total upper bound type operations

Of the three values 0, 1, and 2, 2 is the total upper bound. In Fig. 38.6, there are four operations with the results being the total upper bound 2:

m=f(n_1, 0, n_3, 0)=2

m=f(0, n_2, 0, n_4)=2

m=f(0, n_2, n_3, 0)=2

m=f(n_1, 0, 0, n_4)=2.

Let us investigate m=f(n_1, 0, n_3, 0)=2. It is denoted by the top-left corner plane matrix. It has two partial derivatives:

$$\frac{\partial f}{\partial n1}\Big|_{n2=0,\ n4=0}=0 \text{ and } \frac{\partial f}{\partial n3}\Big|_{n2=0,\ n4=0}=0.$$

38.2.2.4 Compl type operations

In Fig. 38.6, there are 8 compl type operations:

m=f(n_1, 2, n_3, 0)=compl(n_1)

m=f(n_1, 0, n_3, 2)=compl(n_3)

m=f(2, n_2, n_3, 0)=compl(n_2)

m=f(n_1, 0, 2, n_4)=compl(n_4)

m=f(0, 2, n_3, n_4)=compl(n_4)

m=f(2, n_2, 0, n_4)=compl(n_2)

m=f(n_1, 2, 0, n_4)=compl(n_1)

m=f(0, n_2, n_3, 2)=compl(n_3).

Let us investigate m=f(n_1, 2, n_3, 0)=compl(n_1). It is denoted by the bottom-left corner plane matrix of Fig. 38.6. It has two partial derivatives:

$$\frac{\partial f}{\partial n1}\Big|_{n2=2,\ n4=0}=-1 \text{ and } \frac{\partial f}{\partial n3}\Big|_{n2=2,\ n4=0}=0.$$

38.2.2.5 Max-compl type operations

In Fig. 38.6, there are 4 max-compl type operations, which are total ordering based NOR gates:

m=f(n_1, 2, n_3, 2)=max-compl(n_1, n_3)

m=f(2, n_2, 2, n_4)=max-compl(n_2, n_4)

m=f(2, n_2, n_3, 2)=max-compl(n_2, n_3)

m=f(n_1, 2, 2, n_4)=max-compl(n_1, n_4).

Let us investigate m=f(n_1, 2, n_3, 2)=max-compl(n_1, n_3). It is denoted by the bottom-right corner plane matrix of Fig. 38.6. The plane matrix can be regarded as an operation table, the original form of which is the mutually inverse diagram shown in Fig. 38.7. The operation table is obtained by crushing the mutually inverse diagram. Fig. 38.7 is the dual diagram of Fig. 38.1, they can be studied dually. Fig. 38.7 has 5 transaxis straight lines A, B, C, D, and E. A, B, C, and D denote 4 partial derivatives, E denotes a directional derivative. They reveal 5 algebraic properties. Fig. 38.7 has two sums of vectors: $\vec{E}=\vec{A}+\vec{C}$ and $\vec{E}=\vec{B}+\vec{D}$.

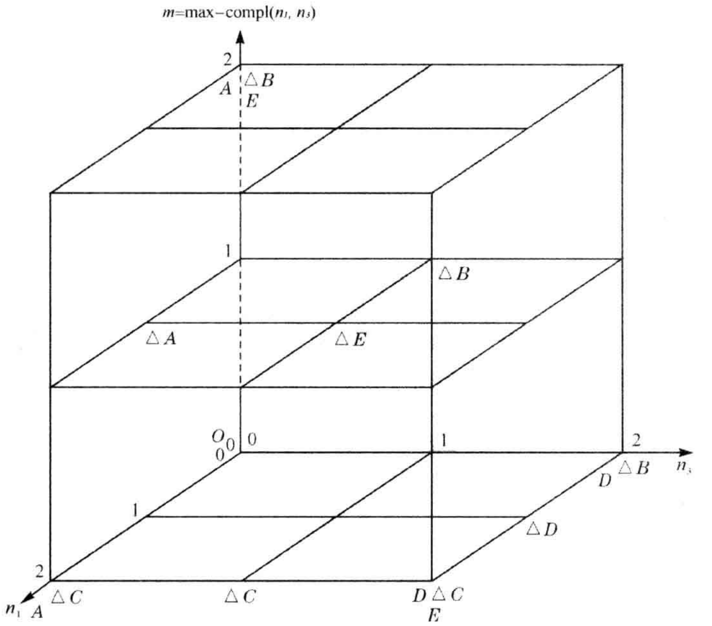


Fig. 38.7 Mutually inverse diagram for m=max-compl(n_1, n_3)

38.2.2.6 Three values fixed, the derivative of f to the fourth value

What Sections 38.2.2.2 through 38.2.2.4 discuss are actually two values fixed, the third value being arbitrary, the partial derivatives of f to the fourth value. The 4 partial derivatives A, B, C, and D discussed in Section 38.2.2.5 are actually three values fixed, the derivatives of f to the fourth value. There are more partial derivatives of the latter case. For example, the middle column of the right-middle plane matrix is a partial derivative $\frac{\partial f}{\partial n1}\Big|_{n2=1,\ n3=1,\ n4=2}=0$.

38.2.2.7 Directional derivatives

The directional derivative E in Section 38.2.2.5 is actually two values fixed, the derivatives of f to the other two values. There are more directional derivatives like this. For example, every plane matrix at the bottom of Fig. 38.6 and every plane matrix at the right of Fig. 38.6 are a directional derivative.

38.2.3 GCD-LCM-compl gates

GCD-LCM-compl gate is a partial ordering based n-valued ANDORN gate. N adopts 2^i (i is any positive integer). We adopt $n=2^2=4$; i.e., the four values 1, 2, 3, and 6, which are the factors of 6. The partial ordering is dividing evenly relation. The carrying set is {1, 2, 3, 6}. The operations GCD (greatest common divisor), LCM (least common multiplier), and compl (complement) make a Boolean algebra to the dividing evenly relation. The complement operation is: 1'=6, 2'=3, 3'=2, 6'=1. We take GCD as the AND operation, take LCM as the OR operation, take compl as the NOT operation, thus, GCD-LCM-compl

constitutes a partial ordering based ANDORN gate. The symbol of GCD-LCM-compl gate is shown in Fig. 38.8, the universal matrix representation of GCD-LCM-compl gate is shown in Fig. 38.9.

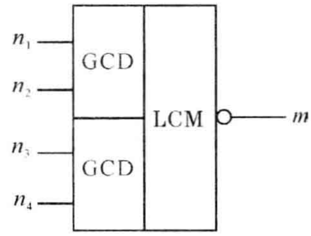


Fig. 38.8 Symbol for GCD-LCM-compl gate

$n_4=$ 1, 2, 3, 6 (left to right); $n_2=$ 1, 2, 3, 6 (top to bottom); n_1 (rows), $n_3=$ (columns)

$n_2 = 1$:

$$\begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&6&6\\6&6&6&6\\6&6&6&6\\6&6&6&6\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&6&3\\6&3&6&3\\6&3&6&3\\6&3&6&3\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&2&2\\6&6&2&2\\6&6&2&2\\6&6&2&2\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&2&1\\6&3&2&1\\6&3&2&1\\6&3&2&1\end{bmatrix}\end{array}$$

$n_2 = 2$:

$$\begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&6&6\\3&3&3&3\\6&6&6&6\\3&3&3&3\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&6&3\\3&3&3&3\\6&3&6&3\\3&3&3&3\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&2&2\\3&3&1&1\\6&6&2&2\\3&3&1&1\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&2&1\\3&3&1&1\\6&3&2&1\\3&3&1&1\end{bmatrix}\end{array}$$

$n_2 = 3$:

$$\begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&6&6\\6&6&6&6\\2&2&2&2\\2&2&2&2\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&6&3\\6&3&6&3\\2&1&2&1\\2&1&2&1\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&2&2\\6&6&2&2\\2&2&2&2\\2&2&2&2\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&2&1\\6&3&2&1\\2&1&2&1\\2&1&2&1\end{bmatrix}\end{array}$$

$n_2 = 6$:

$$\begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&6&6\\3&3&3&3\\2&2&2&2\\1&1&1&1\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&6&3\\3&3&3&3\\2&1&2&1\\1&1&1&1\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&6&2&2\\3&3&1&1\\2&2&2&2\\1&1&1&1\end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{cccc}1&2&3&6\end{array}\\ \begin{array}{c}1\\2\\3\\6\end{array} & \begin{bmatrix}6&3&2&1\\3&3&1&1\\2&1&2&1\\1&1&1&1\end{bmatrix}\end{array}$$

Fig. 38.9 Universal matrix representation of GCD-LCM-compl gate

Concerning the case of two values fixed, the third value being arbitrary, the partial derivative of f to the fourth value, GCD-LCM-compl gate is the same as min-max-compo gate. Therefore, we only investigate the case of three values fixed, the partial derivative of f to the fourth value and the case of directional derivative.

First, let us investigate the plane matrix at the bottom-right corner of Fig. 38.9. It can be viewed as an operation table, denoting LCM-compl(n_1, n_3), partial ordering based NOR gate. The original form of the operation table is the mutually inverse diagram shown in Fig. 38.10. The operation table is obtained by crushing the mutually inverse diagram. In Fig. 38.10, there are 6 transaxis straight lines. The transaxis straight lines marked A, B, C, D,

and F denote three values fixed, the partial derivative of f to the fourth value, in which F is absent in Fig. 38.7. The transaxis straight line E is directional derivative. Fig. 38.10 has three sum of vectors, in which $\vec{F}=\vec{C}+\vec{D}$ is absent in Fig. 38.7.

In addition to Fig. 38.10 having one more three values fixed partial derivative than Fig. 38.7, Fig. 38.9 has more three values fixed partial derivatives than Fig. 38.6 elsewhere. For example, the partial derivative $\frac{\partial f}{\partial n1}\Big|_{n2=1,\ n3=1,\ n4=2}=0$ in min-max-compl gate is one that n_2 and n_3 adopt the middle value , n_4 adopts the total upper bound, the derivative is f to n_1. As to GCD-LCM-compl gate, because there are two middle values: 2 and 3, there are two like partial derivatives. One is $\frac{\partial f}{\partial n1}\Big|_{n2=2,\ n3=2,\ n4=6}=0$, see the second column of the second plane matrix on the right. The other is $\frac{\partial f}{\partial n1}\Big|_{n2=3,\ n3=3,\ n4=6}=0$, see the third column of the third plane matrix on the right.

Fig. 38.9 has more directional derivatives than Fig. 38.6. The latter has 5 directional derivatives, while the former has 9.

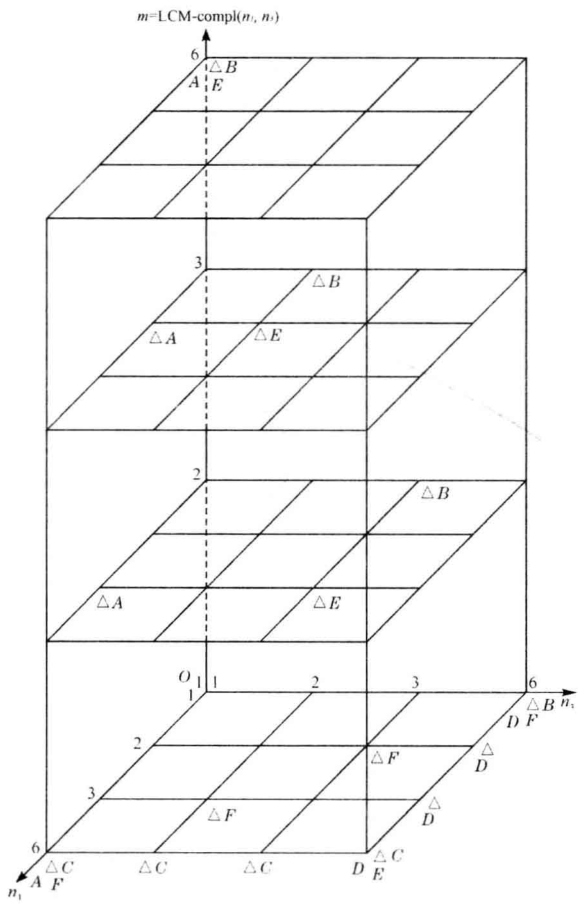


Fig. 38.10 Mutually inverse diagram for m=LCM-compl(n_1, n_3)

38.2.4 Comparison between min-max-compl gate and GCD-LCM-compl gate

The advantages of min-max-compl gate over GCD-LCM-compl gate are: min-max-comple gate is based on total ordering, its values adopted are natural; it can be three-valued logic, the most easily implementable many-valued logic. The advantage of GCD-LCM-compl gate over min-max-compl gate is: GCD-LCM-compl embraces more information: one more sum of vectors, four more directional derivatives, and several more three values fixed partial derivatives.

Someone might say: "You compare 3-valued min-max-compl gate with 4-valued GCD-LCM-compl gate. You should compare 4-valued min-max-compl gate with 4-valued GCD-LCM-compl gate." Our answer is: "(1) If we use 4-valued min-max-compl gate to compare, we will lose the advantage that 3-valued min-max-compl gate is most easily implementable. (2) Even if we use 4-valued min-max-compl gate to compare, 4-valued GCD-LCM-compl gate still has one more partial derivative marked F, one more sum of vectors, two more directional derivatives.

Chapter 39 Applications of mutually-inversistic mathematical analysis

Section 39.1 concerns the application of single-sided discrete calculus to modern control theory. Section 39.2 concerns the application of single-sided discrete calculus and unified calculus to experiment or observation data. Section 39.3 concerns mutually-inversistic fuzzy controller.

39.1 Mutually-inversistic modern control theory

39.1.1 Introduction to mutually-inversistic modern control theory

Traditional modern control theory uses continuous state equations and discrete state equations to describe continuous-time control systems and discrete-time control systems respectively. Mutually-inversistic modern control theory is different from traditional modern control theory. It is based on single-sided discrete calculus, described by mutually-inversistic state equations.

39.1.2 Establishment of mutually-inversistic state equations

39.1.2.1 Establishment of mutually-inversistic state equations from single-sided discrete differential equations

Continuous-time control systems are described by continuous state equations, which can be deduced from differential equations. Suppose the differential equation of a continuous-time control system is

$$y^{(n)}+a_{n-1}y^{(n-1)}+\ldots+a_1\dot{y}+a_0y=b_0u \tag{39.1}$$

From (39.1) we can construct the state variable diagram of the system shown in Fig. 39.1.

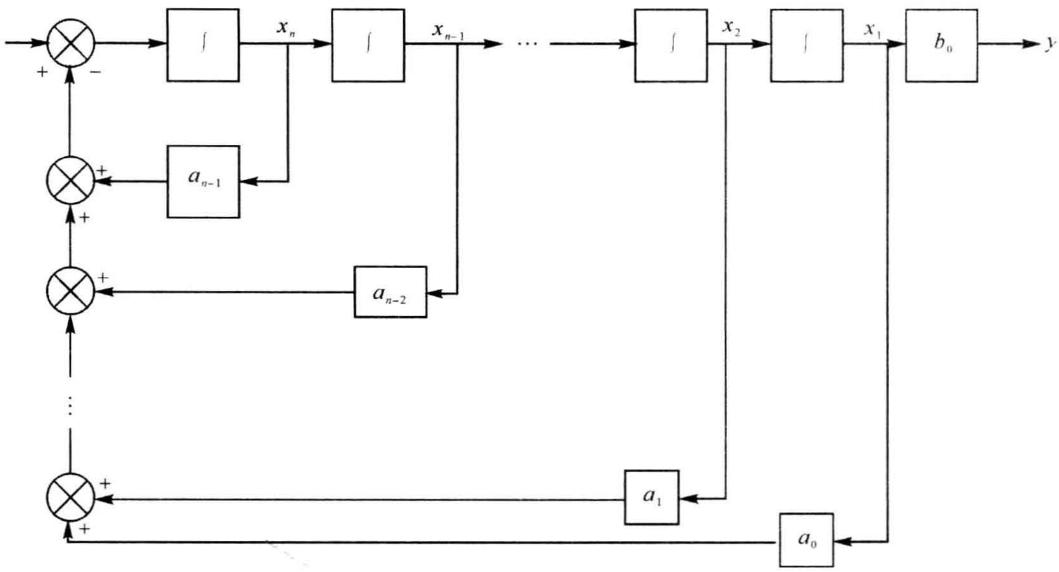


Fig. 39.1 State variable diagram of the differential equation

Take the outputs of every integrator of Fig. 39.1 as state variables, take the inputs as the derivatives of the state variables, we can obtain the following continuous state equations:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\cdots\cdots$$

$$\dot{x}_{n-1} = x_n$$

$$\dot{x}_n = -a_0x_1 - a_1x_2 - \cdots - a_{n-2}x_{n-1} - a_{n-1}x_n + u$$

and the output equation:

$$y = b_0x_1$$

Mutually-inversistic state equations can be established similarly. The first method for the establishment of the mutually-inversistic state equations is the deduction from the single-sided discrete differential equations. Suppose the single-sided discrete differential equation of a discrete system is

$$\frac{\nabla^m f}{\nabla n^m} + a_{m-1}\frac{\nabla^{m-1} f}{\nabla n^{m-1}} + \cdots + a_1\frac{\nabla f}{\nabla n} + a_0 f = b_0 u \tag{39.2}$$

From (39.2) we can construct the state variable diagram of the system shown in Fig. 39.2. $(\nabla n=1)$I in Fig. 39.2 is a single-sided discrete integrator. Take the outputs of every single-sided discrete integrator of Fig. 39.2 as the state variables, take the inputs as the

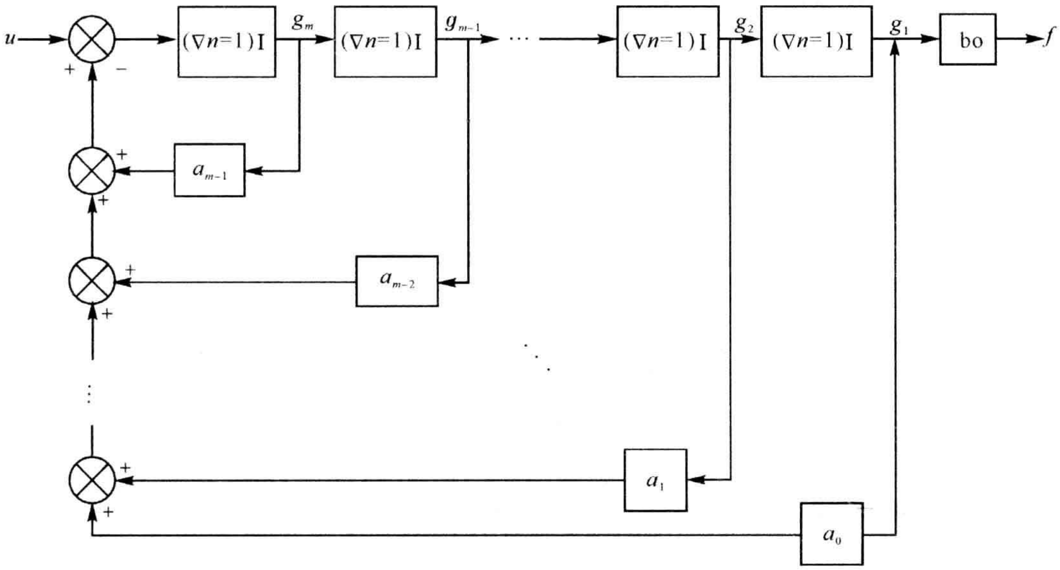


Fig. 39.2 State variable diagram for single-sided discrete differential equation

derivatives of the state variables, we obtain the following mutually-inversistic state equations:

$$\frac{\nabla g_1}{\nabla n}=g_2$$

$$\frac{\nabla g_2}{\nabla n}=g_3$$

……

$$\frac{\nabla g_{m-1}}{\nabla n}=g_m$$

$$\frac{\nabla g_m}{\nabla n}=-a_0g_1-a_1g_2-\cdots-a_{m-2}g_{m-1}-a_{m-1}g_m+u$$

and the mutually-inversistic output equation:

$$f=b_0g_1$$

Written as matrix form, they become

$$\begin{bmatrix}\frac{\nabla g_1}{\nabla n}\\ \frac{\nabla g_2}{\nabla n}\\ \vdots\\ \frac{\nabla g_{m-1}}{\nabla n}\\ \frac{\nabla g_m}{\nabla n}\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ -a_0 & -a_1 & -a_2 & \cdots & -a_{m-1}\end{bmatrix}\begin{bmatrix}g_1\\ g_2\\ \vdots\\ g_{m-1}\\ g_m\end{bmatrix}+\begin{bmatrix}0\\ 0\\ \vdots\\ 0\\ 1\end{bmatrix}u \tag{39.3}$$

$$f=\begin{bmatrix} b_0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_m \end{bmatrix} \quad (39.4)$$

They are abbreviated as

$$\frac{\nabla g}{\nabla n}=Ag+Bu \quad (39.5)$$

$$f=C^T g \quad (39.6)$$

39.1.2.2 Establishment of mutually-inversistic state equations by discretization of continuous state equations

Continuous state equations is a simultaneous first-order differential equations in the form of

$$y'_i=f_i(x_1, y_1, y_2,\ldots,y_m)\ (i=1, 2,\ldots, m) \quad (39.7)$$

Formula (39.7) can be written as

$$\lim_{h\to o}\frac{y_i(x+h)-y_i(x)}{h}=f_i(x_1, y_1, y_2,\ldots,y_m)\ (i=1, 2,\ldots, m) \quad (39.8)$$

Eliminate $\lim_{h\to o}$ in (39.8), we obtain

$$\frac{y_i(x+h)-y_i(x)}{h}=f_i(x_1, y_1, y_2,\ldots,y_m)\ (i=1, 2, \ldots, m) \quad (39.9)$$

When h=1, (39.9) is just mutually-inversistic state equations. So, mutually-inversistic state equations correspond directly to continuous state equations. While discrete state equations can only be obtained by further transformation to (39.9). For example, the discrete state equations:

$$y_i(x+h)=y_i(x)+hf_i(x_1, y_1, y_2,\ldots,y_m)\ (i=1, 2,\ldots, m) \quad (39.10)$$

is obtained by applying Euler method to (39.9). So, discrete state equations do not correspond to continuous state equations directly.

39.1.3 Solving mutually-inversistic state equations

Suppose the mutually-inversistic state equation is

$$\frac{\nabla g}{\nabla n}=Ag+Bu \quad (39.11)$$

Suppose

$$g=b_0+b_1n+b_2n^2,\ u=c \quad (39.12)$$

(The solution of g with higher exponent is similar to this, only more complicated than this.) Substituting (39.12) into (39.11), and noticing $\frac{\nabla n^2}{\nabla n}=2n-1$, we obtain

$$b_1+b_2(2n-1)=A(b_0+b_1n+b_2n^2)+Bc \tag{39.13}$$

Equaling the coefficients of like exponent on both sides of (39.13), we obtain

$$b_1-b_2=Ab_0+Bc \tag{39.14}$$

$$2b_2=Ab_1 \tag{39.15}$$

$$0=Ab_2 \tag{39.16}$$

Formula (39.14) is multiplied by 2, then

$$2b_1-2b_2=2Ab_0+2Bc \tag{39.17}$$

is obtained. Formula (39.15) adds (39.17), obtaining

$$2b_1=Ab_1+2Ab_0+2Bc \tag{39.18}$$

Rearrange (39.18), obtaining

$$(2I-A)b_1=2Ab_0+2Bc \tag{39.19}$$

and

$$b_1=2(2I-A)^{-1}(Ab_0+Bc) \tag{39.20}$$

where I is the identity matrix. Substituting (39.20) into (39.14), we obtain

$$\begin{aligned} b_2&=b_1-Ab_0-Bc\\ &=2(2I-A)^{-1}(Ab_0+Bc)-Ab_0-Bc\\ &=[2(2I-A)^{-1}-I]Ab_0+[2(2I-A)^{-1}-I]Bc \end{aligned} \tag{39.21}$$

Substituting (39.20) and (39.21) into (39.12), we obtain

$$\begin{aligned} g&=b_0+b_1n+b_2n^2\\ &=b_0+2(2I-A)^{-1}(Ab_0+Bc)n+[2(2I-A)^{-1}-I](Ab_0+Bc)n^2\\ &=\{I+2(2I-A)^{-1}An+[2(2I-A)^{-1}-I]An^2\}b_0+\{2(2I-A)^{-1}n+[2(2I-A)^{-1}-I]n^2\}Bc \end{aligned} \tag{39.22}$$

Let n=1, then (39.22) becomes

$$\begin{aligned} g(1)&=b_0+2(2I-A)^{-1}(Ab_0+Bc)+[2(2I-A)^{-1}-I](Ab_0+Bc)\\ &=\{I+2(2I-A)^{-1}A+[2(2I-A)^{-1}-I]A\}b_0+\{2(2I-A)^{-1}+[2(2I-A)^{-1}-I]\}Bc \end{aligned} \tag{39.23}$$

Rearrange (39.23), we obtain

$$b_0=\{I+2(2I-A)^{-1}A+[2(2I-A)^{-1}-I]A\}^{-1}\{g(1)-\{2(2I-A)^{-1}+[2(2I-A)^{-1}-I]\}Bc\} \tag{39.24}$$

Substituting (39.24) into (39.22), we obtain

$$g=\{I+2(2I-A)^{-1}An+[2(2I-A)^{-1}-I]An^2\}\{I+2(2I-A)^{-1}A+[2(2I-A)^{-1}-I]A\}^{-1}\{g(1)-\{2(2I-A)^{-1}+[2(2I-A)^{-1}-I]\}Bc\}+\{2(2I-A)^{-1}n+[2(2I-A)^{-1}-I]n^2\}Bc \tag{39.25}$$

Formula (39.25) is the solution of the mutually-inversistic state equation (39.11).

39.1.4 Controllability and observability

The controllability and observability of the discrete-time system denoted by mutually-inversistic state equations are the same as those of the continuous-time system.

39.1.4.1 Controllability

39.1.4.1.1 Controllability of Jordan canonical systems

Suppose the mutually-inversistic state equation of the system is

$$\frac{\nabla g}{\nabla n}=Ag+Bu \tag{39.26}$$

Let g=Th, then (39.26) can be transformed into Jordan canonical form

$$\frac{\nabla h}{\nabla n}=\Lambda h+T^{1}Bu \tag{39.27}$$

or

$$\frac{\nabla h}{\nabla n}=Jh+T^{1}Bu \tag{39.28}$$

where

$$\Lambda=\begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \lambda_3 & & \\ & & & \vdots & \\ & & & & \lambda_m \end{bmatrix} \tag{39.29}$$

$\lambda_1 \neq \lambda_2 \neq \lambda_3 \neq \cdots \neq \lambda_m$, and

$$\mathbf{J}=\begin{bmatrix} \lambda_1 & 1 & & & 0 \\ & \lambda_2 & 1 & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & \cdot & 1 \\ 0 & & & & \lambda_1 \end{bmatrix} \tag{39.30}$$

λ_1 is a multiple m eigenvalue.

The controllability criteria of the system is

(1) If the eigenvalues of the system matrix A are all distinct, then (39.26) can be transformed into (39.27), at this time, the sufficient and necessary condition for the controllability of the system is that the control matrix $T^{-1}B$ has no row which is all zeros.

(2) If all the eigenvalues of the system matrix A are the same, then (39.26) can be transformed into (39.28), at this time, the sufficient and necessary condition for the

controllability of the system is that the last row of the control matrix $T^{1}B$ is not all zeros.

39.1.4.1.2 Determination of system controllability from A and B directly

The mutually-inversistic state equation is the same as (39.26). The sufficient and necessary condition for the system controllability is that the rank of the controllability matrix

$$M=[B\ \ AB\ \ A_2B\ \ \cdots\ \ A^{m-1}B] \tag{39.31}$$

is m.

39.1.4.2 Observability

39.1.4.2.1 Observability of Jordan canonical systems

Suppose the mutually-inversistic state equation of the system is

$$\frac{\nabla g}{\nabla n}=Ag \quad g(n_0)=g_0 \tag{39.32}$$

The mutually-inversistic output equation is

$$f=Cg \tag{39.33}$$

Let g=Th, then (39.32) and (39.33) can be transformed into Jordan canonical form

$$\frac{\nabla h}{\nabla n}=\Lambda h \tag{39.34}$$

$$f=CTh \tag{39.35}$$

or

$$\frac{\nabla h}{\nabla n}=Jh \tag{39.36}$$

$$f=CTh \tag{39.37}$$

The criteria of system observability are

(1) If the eigenvalues of the system matrix A are all distinct, then (39.32) and (39.33) can be transformed into (39.34) and (39.35), at this time, the sufficient and necessary condition for system observability is that the output matrix CT has no column which is all zeros.

(2) If the eigenvalues of the system matrix A are all the same, then (39.32) and (39.33) can be transformed into (39.36) and (39.37), at this time, the sufficient and necessary condition for system observability is that the first column of the output matrix CT is not all zeros.

39.1.4.2.2 Determination of system observability from A and C directly

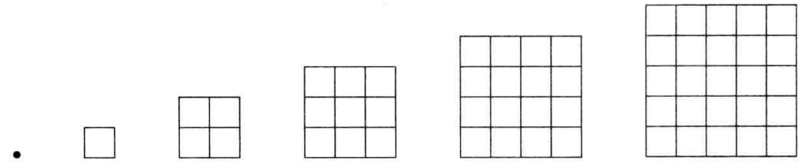
The mutually-inversistic state equation and output equation are the same as (39.32) and (39.33). The sufficient and necessary condition for the system observability is that the rank of the observability matrix

$$N=\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{m-1} \end{bmatrix} \tag{39.38}$$

is m.

39.1.5 An example

Example 39.1: Suppose we have a series of squares shown in Fig. 39.3, where the sides of the squares are natural numbers.



n	0	1	2	3	4	5
$f=n^2$	0	1	4	9	16	25
$\frac{\nabla f}{\nabla n}$		1	3	5	7	9
$\nabla^2 f/\nabla n^2$			2	2	2	2

Fig. 39.3 Figure for Example 39.1

In Fig. 39.3, the first row below the series of squares is the sides n of the squares, the second row is the areas $f=n^2$ of the squares, the third row is $\frac{\nabla f}{\nabla n}$, the fourth row is $\frac{\nabla^2 f}{\nabla n^2}$.

Based on Fig. 39.3, we can establish the initial value problem of the single-sided discrete differential equation:

$$\frac{\nabla^2 f}{\nabla n^2}=2,\quad f(1)=1,\quad \frac{\nabla f}{\nabla n}\Big|_{n=1}=1. \tag{39.39}$$

In (39.39), suppose $g_1=f$, then $g_2=\frac{\nabla g_1}{\nabla n}=\frac{\nabla f}{\nabla n}$. We obtain the mutually-inversistic state equation:

$$\begin{bmatrix} \frac{\nabla g_1}{\nabla n} \\ \frac{\nabla g_2}{\nabla n} \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}2 \quad \begin{bmatrix} g_1(1) \\ g_2(1) \end{bmatrix}=\begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{39.40}$$

and the mutually-inversistic output equation:

$$f=[1\ \ 0]\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \tag{39.41}$$

From (39.40), we know

$$A=\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{39.42}$$

$$B=\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{39.43}$$

$$\begin{bmatrix} g_1(1) \\ g_2(1) \end{bmatrix}=\begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{39.44}$$

From (39.41), we know

$$C=[1 \quad 0] \tag{39.45}$$

Substitute (39.42), (39.43), (39.44), and u=c=2 into (39.25), we obtain the solution of (39.40):

$$\begin{bmatrix} g_1 \\ g_2 \end{bmatrix}=\begin{bmatrix} n^2 \\ 2n-1 \end{bmatrix} \tag{39.46}$$

From (39.46), we know $g_1=f=n^2$ and $g_2=\dfrac{\nabla f}{\nabla n}=2n-1$.

The characteristic equation of the system is

$$\lambda^2=0 \tag{39.47}$$

where $\lambda=0$ is a multiple 2 root. Formula (39.40) is already a Jordan canonical form with multiple 2 root $\lambda=0$. From (39.43) we know that the last row of the control matrix B is not 0, hence the system is controllable. From (39.35) we know that the first column of the output matrix C is not 0, hence the system is observable.

39.1.6 Comparison between mutually-inversistic state equations and discrete state equations

If discrete state equation is used to solve Example 39.1, then the solution is

$$x(n)=G^n x(0)\sum_{j=0}^{n-1}G^{n-j-1}Hu(j)=\begin{bmatrix} n^2 \\ n^2+2n+1 \end{bmatrix} \tag{39.48}$$

where x(n) is the state, x(0) is the initial state, G^n is the state transition matrix, Hu(n) is the input.

The advantages of mutually-inversistic state equation over discrete state equation are:

(1) Mutually-inversistic state equation directly corresponds to the discretization of continuous state equation, while discrete state equation does not.

(2) From (39.46) we learn that the solution of the mutually-inversistic state equation is composed of the nth state and its derivatives. While from (39.48) we learn that the solution of the discrete stare equation is composed of the nth and the (n+1)th states. The former is

more informative than the latter.

(3) From (39.25) we learn that the computation of the solution of mutually-inversistic state equation is simple. It only needs the inverse operation of matrix, the product of two matrices, the addition of two matrices. While from (39.48) we learn that when solving the discrete state equation, we need to compute G^n and $\sum_{j=0}^{n-1} G^{n-j-1} Hu(j)$, which are more hard and need skills.

The advantage of discrete state equation over mutually-inversistic state equation is: From (39.48) we learn that the solution of the discrete state equation is simple, general; e.g., it can be used for any input; and has specific meaning; e.g., $G^n x(0)$ is the multiplication of the state transition matrix and the initial state-the impact of the initial state, $\sum_{j=0}^{n-1} G^{n-j-1} Hu(j)$ is the convolution of the state transition matrix and the input-the impact of the input. While from (39.25) we learn that the solution of the mutually-inversistic state equation is complicated, not general; e.g., when the input is a exponential function, then (39.25) is no longer applicable; and does not have specific meaning; e.g., we cannot point out what function each part of the formula has.

39.2 Application of single-sided discrete calculus and unified calculus to experiment or observation data

39.2.1 Introduction

In Chapters 20 and 21, the functions dealt with by single-sided discrete calculus and unified calculus are analytic functions, such as n^2, 2^n. Experiment or observation data are irregular. Up to now, these data are dealt with by numerical analysis, which construct interpolation functions for them, and make numerical differentiation and integration. In this section, experiment or observation data are viewed as arbitrary single-sided discrete function or unified function, which are dealt with by single-sided discrete calculus and unified calculus directly without interpolation functions. This method is simple, informative, and sound in conclusion.

39.2.2 Single-sided discrete calculus of arbitrary single-sided discrete functions

39.2.2.1 Derivatives

Suppose we have an arbitrary single-sided discrete function f(n) shown in Table 39.1.

Table 39.1 An arbitrary single-sided discrete function f(n)

n	0	1	2	3	4	5	6	7
f(n)	2	5	4	3	1	6	6	7

The diagram for f(n) is shown in Fig. 39.4.

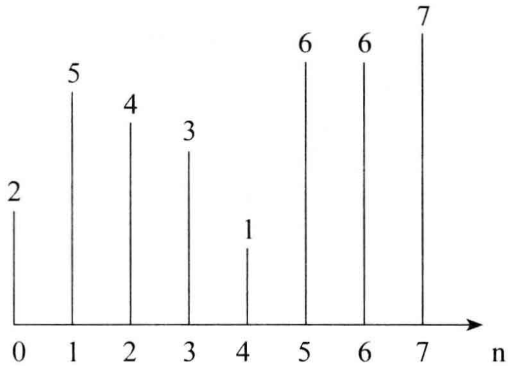


Fig. 39.4 Diagram for the arbitrary single-sided discrete function f(n)

The process from f(n) to $\frac{\nabla f}{\nabla n}$ sequence is as follows: the 0^{th} term of $\frac{\nabla f}{\nabla n}$ sequence is f(0)=2. The 1^{st} term of $\frac{\nabla f}{\nabla n}$ sequence is $\frac{\nabla f(1)}{\nabla n}$ =f(1) − f(0)=5 − 2=3. The 2^{nd} term of $\frac{\nabla f}{\nabla n}$ sequence is $\frac{\nabla f(2)}{\nabla n}$=f(2) − f(1)=4 − 5= − 1. The rest can be inferred by analogy. The $\frac{\nabla f}{\nabla n}$ sequence obtained is shown in Table 39.2.

Table 39.2 $\frac{\nabla f}{\nabla n}$ sequence

n	0	1	2	3	4	5	6	7
$\frac{\nabla f}{\nabla n}$ sequence	2	3	− 1	− 1	− 2	5	0	1

We call it $\frac{\nabla f}{\nabla n}$ sequence not $\frac{\nabla f}{\nabla n}$ function, because the 0^{th} term of it is f(0) not a first-order derivative.

If f(i+1) > f(i), then $\frac{\nabla f(i+1)}{\nabla n}$ > 0. For example, f(1)=5 > f(0)=2, so $\frac{\nabla f(1)}{\nabla n}$=3 > 0. If f(i+1) < f(i), then $\frac{\nabla f(i+1)}{\nabla n}$ < 0. For example, f(2)=4 < f(1)=5, so $\frac{\nabla f(2)}{\nabla n}$ = − 1 < 0. These are consistent with Newtonian calculus. But $\frac{\nabla f(i+1)}{\nabla n}$=0 does not necessarily mean that f(i+1) is an extremum, it only means f(i)=f(i+1). This is not fully consistent with Newtonian

calculus. For example, f(5)=f(6)=6, hence $\frac{\nabla f(6)}{\nabla n}$=0, but f(6) is not an extremum.

The process from $\frac{\nabla f}{\nabla n}$ sequence to $\frac{\nabla^2 f}{\nabla n^2}$ sequence is as follows: the 0th term of the $\frac{\nabla^2 f}{\nabla n^2}$ sequence is the 0th term of $\frac{\nabla f}{\nabla n}$ sequence f(0)=2. The 1st term of the $\frac{\nabla^2 f}{\nabla n^2}$ sequence is the 1st term of the $\frac{\nabla f}{\nabla n}$ sequence minus the 0th term of the $\frac{\nabla f}{\nabla n}$ sequence $\frac{\nabla f(1)}{\nabla n} - f(0)=3-2=1$. The 2nd term of the $\frac{\nabla^2 f}{\nabla n^2}$ sequence is $\frac{\nabla^2 f(2)}{\nabla n^2}=\frac{\nabla f(2)}{\nabla n}-\frac{\nabla f(1)}{\nabla n}=-1-3=-4$. The 3rd term of the $\frac{\nabla^2 f}{\nabla n^2}$ sequence is $\frac{\nabla^2 f(3)}{\nabla n^2}=\frac{\nabla f(2)}{\nabla n}-\frac{\nabla f(2)}{\nabla n}=-1-(-1)=0$. The rest can be inferred by analogy. The $\frac{\nabla^2 f}{\nabla n^2}$ sequence constructed is shown in Table 39.3.

Table 39.3 $\frac{\nabla^2 f}{\nabla n^2}$ **sequence**

n	0	1	2	3	4	5	6	7
$\frac{\nabla^2 f}{\nabla n^2}$ sequence	2	1	−4	0	−1	7	−5	1

We call it $\frac{\nabla^2 f}{\nabla n^2}$ sequence, not $\frac{\nabla^2 f}{\nabla n^2}$ function, because the 0th term f(0) and the 1st term $\frac{\nabla f(1)}{\nabla n}$ − f(0) of it are not second order derivatives.

When f(i), f(i+1), and f(i+2) present themselves as a concave figure, then $\frac{\nabla^2 f(i+2)}{\nabla n^2} >$ 0. For example, f(5), f(6), and f(7) present themselves as a concave figure, therefore, $\frac{\nabla^2 f(7)}{\nabla n^2}$ =1 > 0. When f(i), f(i+1), and f(i+2) present themselves as a convex figure, then $\frac{\nabla^2 f(i+2)}{\nabla n^2}$ < 0. For example, f(0), f(1), and f(2) present themselves as a convex figure, therefore, $\frac{\nabla^2 f(2)}{\nabla n^2}=-4<0$. These are consistent with Newtonian calculus. But $\frac{\nabla^2 f(i+2)}{\nabla n^2}$=0 does not necessarily means f(i+1) is a inflection point, it only means f(i), f(i+1), and f(i+2) are on the same straight line. This is not fully consistent with Newtonian calculus. For example, f(1), f(2), and f(3) are on the same straight line, therefore, $\frac{\nabla^2 f(3)}{\nabla n^2}$=0, but f(2) is not a inflection point.

39.2.2.2 Integral

Suppose we know $\frac{\nabla^2 f}{\nabla n^2}$ and we want to find the primitive function f(n), what we do

is to solve the single-sided discrete ordinary differential equation. At this time, we need to know not only $\frac{\nabla^2 f}{\nabla n^2}$ but also the initial condition f(0) and $\frac{\nabla f(1)}{\nabla n}$. They form the $\frac{\nabla^2 f}{\nabla n^2}$ sequence, shown in Table 39.3. We make single-sided discrete integral on the $\frac{\nabla^2 f}{\nabla n^2}$ sequence, obtaining the $\frac{\nabla f}{\nabla n}$ sequence, shown in Table 39.2. Then, we make single-sided discrete integral on the $\frac{\nabla f}{\nabla n}$ sequence, obtaining the primitive function f(n), shown in Table 39.1.

The process of making single-sided discrete integral on Table 39.3 to obtain Table 39.2 is as follows: we take the 0^{th} term f(0)=2 of Table 39.3 as the 0^{th} term of Table. 39.2. we take the sum of the 0^{th} and 1^{st} terms of Table 39.3 as the 1^{st} term of Table 39.2. Because the 0^{th} term of Table 39.3 is just the 0^{th} term of Table 39.2, what we actually do is to add the 0^{th} term f(0)=2 of Table 39.2 and the 1^{st} term $\frac{\nabla f(1)}{\nabla n} - f(0)=1$ of Table 39.3 to obtain the 1^{st} term $\frac{\nabla f(1)}{\nabla n}=3$ of Table 39.2. We take the sum of the 0^{th}, 1^{st}, and 2^{nd} terms of Table 39.3 as the 2^{nd} term of Table 39.2. Because the sum of the 0^{th} and 1^{st} terms of Table 39.3 is the 1^{st} term of Table 39.2, what we actually do is to add the 1^{st} term $\frac{\nabla f(1)}{\nabla n}=3$ of Table 39.2 and the 2^{nd} term $\frac{\nabla^2 f(2)}{\nabla n^2}=\frac{\nabla f(2)}{\nabla n} - \frac{\nabla f(1)}{\nabla n}= -4$ of Table 39.3 to obtain the 2^{nd} term $\frac{\nabla f(2)}{\nabla n}= -1$ of Table 39.2. The rest can be inferred by analogy.

The process of making single-sided discrete integral on Table 39.2 to obtain the primitive function f(n) of Table 39.1 is similar.

39.2.3 Unified calculus of arbitrary unified functions

39.2.3.1 Derivatives

The arbitrary unified function is in the form of $f(n)|_{\nabla n=h}$, where f(n) is a real number, and h is also a real number. When $h\to 0$, it is the special case of the continuous function in Newtonian calculus. When h=1, it is the special case of the single-sided discrete function in single-sided discrete calculus.

Suppose we have an arbitrary unified function $f(n)|_{\nabla n=0.5}$, shown in Table 39.4.

Table 39.4 An arbitrary unified function $f(n)|_{\nabla n=0.5}$

n	0	0.5	1	1.5	2	2.5	3	3.5
$f(n)\|_{\nabla n=0.5}$	2	5	4	3	1	6	6	7

The process from $f(n)|_{\nabla n=0.5}$ to $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence is as follows: the 0^{th} term 2 of

$f(n)|_{\nabla n=0.5}$ is the 0^{th} term of $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence. The 0.5^{th} term 5 of $f(n)|_{\nabla n=0.5}$ minus the 0^{th} term 2 of $f(n)|_{\nabla n=0.5}$ and then divided by 0.5 equaling 6 is the 0.5^{th} term of $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence. The rest can be inferred by analogy. The $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence obtained is shown in Table 39.5.

Table 39.5 $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ **sequence**

n	0	0.5	1	1.5	2	2.5	3	3.5
$\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence	2	6	-2	-2	-4	10	0	2

The 0^{th}, 0.5^{th}, 1^{st},…, terms of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence are $f(0)|_{\nabla n=0.5}$, $\frac{\nabla f(0.5)}{\nabla n}=(f(0.5)|_{\nabla n=0.5}-f(0)|_{\nabla n=0.5})/0.5$, $\frac{\nabla f(1)}{\nabla n}=(f(1)|_{\nabla n=0.5}-f(0.5)|_{\nabla n=0.5})/0.5$,…. The situations revealed by $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}>0$, $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}<0$, and $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}=0$ are the same as those of the $\frac{\nabla f}{\nabla n}$ sequence.

The process from $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence to $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence is as follows: the 0^{th} term 2 of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence is the 0^{th} term of the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence. The 0.5^{th} term 6 of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence minus the 0^{th} term 2 of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence and then divided by 0.5 equaling 8 is the 0.5^{th} term of the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence. The rest can be inferred by analogy. The $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence obtained is shown in Table 39.6.

Table 39.6 $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ **sequence**

n	0	0.5	1	1.5	2	2.5	3	3.5
$\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence	2	8	-16	0	-4	28	-20	4

The 0^{th}, 0.5^{th}, 1^{st}, 1.5^{th} ,…, terms of the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence are $f(0)|_{\nabla n=0.5}$, $(\frac{\nabla f(0.5)}{\nabla n}|_{\nabla n=0.5}-f(0)|_{\nabla n=0.5})/0.5$, $(\frac{\nabla f(1)}{\nabla n}|_{\nabla n=0.5}-\frac{\nabla f(0.5)}{\nabla n}|_{\nabla n=0.5})/0.5$, $(\frac{\nabla f(1.5)}{\nabla n}|_{\nabla n=0.5}-\frac{\nabla f(1)}{\nabla n}|_{\nabla n=0.5})/0.5$,…. The situations revealed by $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}>0$, $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}<0$, and $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}=0$ are the same as those of the $\frac{\nabla^2 f}{\nabla n^2}$ sequence.

39.2.3.2 Integrals

Suppose we know $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ and we want to find the primitive function $f(n)|_{\nabla n=0.5}$, what we do is to solve the unified ordinary differential equation. At this time, we need to know not only $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ but also the initial condition $f(0)|_{\nabla n=0.5}$ and $\frac{\nabla f(0.5)}{\nabla n}|_{\nabla n=0.5}$. They form the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence, shown in Table 39.6. We make unified integral on the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence, obtaining the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence, shown in Table 39.5. Then, we make unified integral on the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence, obtaining the primitive function $f(n)|_{\nabla n=0.5}$, shown in Table 39.4.

The process of making unified integral to Table 39.6 to obtain Table 39.5 is as follows:

the 0^{th} term 2 of the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence is the 0^{th} term of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence. The 0.5^{th} term 8 of the $\frac{\nabla^2 f}{\nabla n^2}|_{\nabla n=0.5}$ sequence times 0.5 plus the 0^{th} term 2 of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence is the 0.5 term of the $\frac{\nabla f}{\nabla n}|_{\nabla n=0.5}$ sequence. The rest can be inferred by analogy.

The process of making unified integral to Table 39.5 to obtain the primitive function $f(n)|_{\nabla n=0.5}$ of Table 39.4 is similar.

39.2.4 Analysis of Chinese GDP during the 11^{th} 5-year plan period

Example 39.2: Table 39.7 is Chinese GDP during the 11^{th} 5-year plan period.

Table 39.7 Chinese GDP during the 11^{th} 5-year plan period (trillion CNY)

n (year)	2005	2006	2007	2008	2009	2010
GDP	18.4937	21.6314	26.5810	31.4045	34.0902	39.7983

Table 39.8 is the normalized Chinese GDP during the 11^{th} 5-year plan period using 18.4937 trillion CNY of the year 2005 as the basis.

Table 39.8 Normalized Chinese GDP during the 11^{th} 5-year plan period

n (year)	2005	2006	2007	2008	2009	2010
GDP	1	1.17	1.44	1.70	1.84	2.15

Fig. 39.5 is the diagram for Table 39.8.

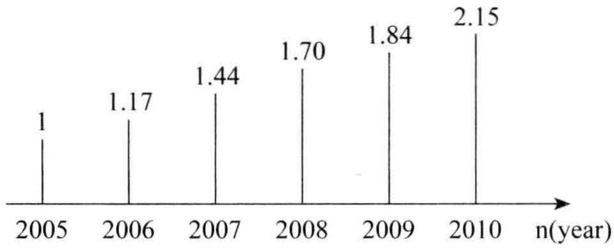


Fig. 39.5 Diagram for Table 39.8

Table 39.9 is the normalized $\frac{\nabla f}{\nabla n}$ sequence of Chinese GDP during the 11th 5-year plan period.

Table 39.9 Normalized $\frac{\nabla f}{\nabla n}$ sequence of Chinese GDP during the 11th 5-year plan period

n (year)	2005	2006	2007	2008	2009	2010
$\frac{\nabla f}{\nabla n}$ sequence	1	0.17	0.27	0.26	0.14	0.31

From Table 39.9 we see that during the 11th 5-year plan period, $\frac{\nabla f}{\nabla n}$ are all greater than 0, reflecting that in this period, Chinese GDP increases year by year. Table 39.10 is the normalized $\frac{\nabla^2 f}{\nabla n^2}$ sequence of Chinese GDP during the 11th 5-year plan period.

Table 39.10 Normalized $\frac{\nabla^2 f}{\nabla n^2}$ sequence of Chinese GDP during the 11th 5-year plan period

n (year)	2005	2006	2007	2008	2009	2010
$\frac{\nabla^2 f}{\nabla n^2}$ sequence	1	− 0.83	0.10	− 0.01	− 0.12	0.17

From Table 30.10 we see that in the years 2008 and 2009, $\frac{\nabla^2 f}{\nabla n^2}$ are less than 0, reflecting that, influenced by the 2008 international financial crisis, in these two years, the increase speed of Chinese GDP decreases. (Note: 2006's − 0.83 is not $\frac{\nabla^2 f}{\nabla n^2}$, but $\frac{\nabla f}{\nabla n}$(2006) − f(2005)).

This example is one of single-sided discrete left derivative on an arbitrary single-sided discrete function.

39.2.5 Personal banking

Example 39.3: A person had a balance of 240 thousand CNY in a bank in 2009. In 2010, he deposited 120 thousand CNY and withdrew 40 thousand CNY. In 2011, he deposited 110 thousand CNY and withdrew 60 thousand CNY. What was his balance in 2010 and 2011?

Solution: Suppose f(n) is the balance. f(2009)=240(thousand CNY). $\frac{\nabla f}{\nabla n}(2010)=120-40=80$(thousand CNY). $\frac{\nabla f}{\nabla n}(2011)=110-60=50$(thousand CNY). The $\frac{\nabla f}{\nabla n}$ sequence is shown in Table 39.11.

Table 39.11 $\frac{\nabla f}{\nabla n}$ **sequence of personal banking (thousand CNY)**

n	2009	2010	2011
$\frac{\nabla f}{\nabla n}$ sequence	240	80	50

We make single-sided discrete integral on Table 39.11, obtaining the balance f shown in Table 39.12.

Table 39.12 Balance f(n)

n	2009	2010	2011
f(n)	240	320	370

From Table 39.12, we know that his balance in 2011 was 320 thousand CNY, and his balance in 2012 was 370 thousand CNY.

This example is one of single-sided discrete integral on an arbitrary single-sided discrete function.

39.3 Mutually-inversistic fuzzy controller

39.3.1 Fuzzy controller of water level

We describe a fuzzy controller of water level. Suppose we have a water container with variable water level and a valve that can pour water into the container and drain water off. Let us design a fuzzy controller that keeps the water level near the 0 point through the valve.

The observation quantities include the deviation of the water level relative to the 0 point e, the first derivative of e with respect to the time interval n $\nabla e/\nabla n$, and the second derivative of e with respect the n $\nabla^2 e/\nabla n^2$.

E is divided into 5 levels and 7 ranks. The 5 levels are negative big (NB, -2), negative small (NS, -1), zero (0), positive small (PS, 1), positive big (PB, 2). The 7 ranks are -3, -2, -1, 0, +1, +2, +3. These determine Table 39.13.

Table 39.13 Membership grade for e

Rank of e / Membership grade / Level of e	−3	−2	−1	0	+1	+2	+3
PBe	0	0	0	0	0	0.5	1
PSe	0	0	0	0	1	0.5	0
0e	0	0	0.5	1	0.5	0	0
NSe	0	0.5	1	0	0	0	0
NBe	1	0.5	0	0	0	0	0

Likewise, $\nabla e/\nabla n$ and $\nabla^2 e/\nabla n^2$ are shown in Tables 39.14 and 39.15 respectively.

Table 39.14 Membership grade for $\nabla e/\nabla n$

Rank of $\nabla e/\nabla n$ / Membership grade / Level of $\nabla e/\nabla n$	−3	−2	−1	0	+1	+2	+3
PB $\nabla e/\nabla n$	0	0	0	0	0	0.5	1
PS $\nabla e/\nabla n$	0	0	0	0	1	0.5	0
0 $\nabla e/\nabla n$	0	0	0.5	1	0.5	0	0
NS $\nabla e/\nabla n$	0	0.5	1	0	0	0	0
NB $\nabla e/\nabla n$	1	0.5	0	0	0	0	0

Table 39.15 Membership grade for $\nabla^2 e/\nabla n^2$

Rank of $\nabla^2 e/\nabla n^2$ / Membership grade / Level of $\nabla^2 e/\nabla n^2$	−3	−2	−1	0	+1	+2	+3
PB $\nabla^2 e/\nabla n^2$	0	0	0	0	0	0.5	1
PS $\nabla^2 e/\nabla n^2$	0	0	0	0	1	0.5	0
0 $\nabla^2 e/\nabla n^2$	0	0	0.5	1	0.5	0	0
NS $\nabla^2 e/\nabla n^2$	0	0.5	1	0	0	0	0
NB $\nabla^2 e/\nabla n^2$	1	0.5	0	0	0	0	0

The control quantity u is the variation of degree of the valve rotation, anti-clockwise rotation is positive (pouring water into), clockwise rotation is negative (drain water off). The variation of degree of the valve rotation is divided into 5 levels and 9 ranks. This determines Table 39.16.

Table 39.16 Membership grade for u

Rank of u / Membership grade / Level of u	−4	−3	−2	−1	0	+1	+2	+3	+4
PBu	0	0	0	0	0	0	0	0.5	1
PSu	0	0	0	0	0	0.5	1	0.5	0
0u	0	0	0	0.5	1	0.5	0	0	0
NSu	0	0.5	1	0.5	0	0	0	0	0
NBu	1	0.5	0	0	0	0	0	0	0

The language control rule is: "if level $(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$, then level u= $-(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$".

Example 39.4: Suppose rank e is 0, rank $\nabla e/\nabla n$ is 1, and rank $\nabla^2 e/\nabla n^2$ is 2, determine rank u.

Solution: Level e is 0, level $\nabla e/\nabla n$ is 1 (PS), level $\nabla^2 e/\nabla n^2$ is 1(PS). Level $(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$ is (0+1+1)/3=0.67, round off to 1. Level u= $-(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$ is −1. Rank u is −2.

From this example we see that because $\nabla e/\nabla n$ and $\nabla^2 e/\nabla n^2$ are greater than 0, even if e=0, we still need to pour into water.

Mutually-inversistic fuzzy controller of water level has three advantages over conventional fuzzy controller of water level. First, in conventional fuzzy controller, only e is considered, while in mutually-inversistic fuzzy controller, in addition to e, $\nabla e/\nabla n$ and $\nabla^2 e/\nabla n^2$ are also considered. Secondly, in conventional fuzzy controller, we need 5 language control rules: "if e NB, then u PB", "if e NS, then u PS", if e=0, then u=0", "if e PS, then u NS', "if e PB, then u NB". While in mutually-inversistic fuzzy controller, only one language control rule is needed: "if level $(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$, then level u= $-(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$". Thirdly, in conventional fuzzy controller, in order to compute u, composition of relational matrices is needed. While in mutually-inversistic fuzzy controller, hypothetical inference is carried out: "if level $(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$, then level u= $-(e+\nabla e/\nabla n+\nabla^2 e/\nabla n^2)/3$" is the major premise, (0+1+1)/3 is the minor premise, u= −1 is the conclusion, much simpler.

39.3.2 Fuzzy controller of missile striking aircraft carrier

Suppose longitude is the x-axis, eastward is the positive direction of the x-axis; latitude is the y-axis, northward is the positive direction of the y-axis. An aircraft carrier navigates from southeast to northwest. The displacement of the aircraft carrier is (x_{AC}, y_{AC}). The velocity of the aircraft carrier is decomposed into $\nabla x_{AC}/\nabla n$ and $\nabla y_{AC}/\nabla n$. A missile goes in opposite direction with the aircraft carrier. Its displacement is (x_M, y_M). Its velocity is decomposed into $\nabla x_M/\nabla n$ and $\nabla y_M/n$, which can be controlled by the graduations of x_M and y_M azimuth control of the azimuth controller of the missile. The velocity of the wind is decomposed into $\nabla x_W/\nabla n$ and $\nabla y_W/\nabla n$.

Suppose the initial displacement of the aircraft carrier is (x_{ACinit}, y_{ACinit}), then the initial point of impact of the missile is (x_{ACinit}, y_{ACinit}), corresponding to which there are the initial graduations of x_M and y_M azimuth control.

After the missile is launched, the course and velocity of the aircraft carrier change constantly, the direction and velocity of the wind also change constantly. Therefore, fuzzy control rules should be used to adjust constantly the graduation of the azimuth controller of the missile to correct the point of impact. The fuzzy control rule of the x-axis is:

$\neg_f((\nabla x_{AC}/\nabla n)\text{fast})\wedge_f((\nabla x_W/\nabla n)\text{strong})\leqslant_f^{-1}\neg_f(($deviation from the initial graduation of x_M azimuth control of the missile) big)

(39.49)

The fuzzy control rule of the y-axis is similar. In (39.49), "fast", "strong", and "big" are fuzzy concepts.

Suppose the scope of $\nabla x_{AC}/\nabla n$ is $((\nabla x_{AC}/\nabla n)_{min}, (\nabla x_{AC}/\nabla n)_{max})=(-30$ sea miles/h, +30 sea miles/h), then the membership function of "($\nabla x_{AC}/\nabla n$)fast" is shown in Fig. 39.6.

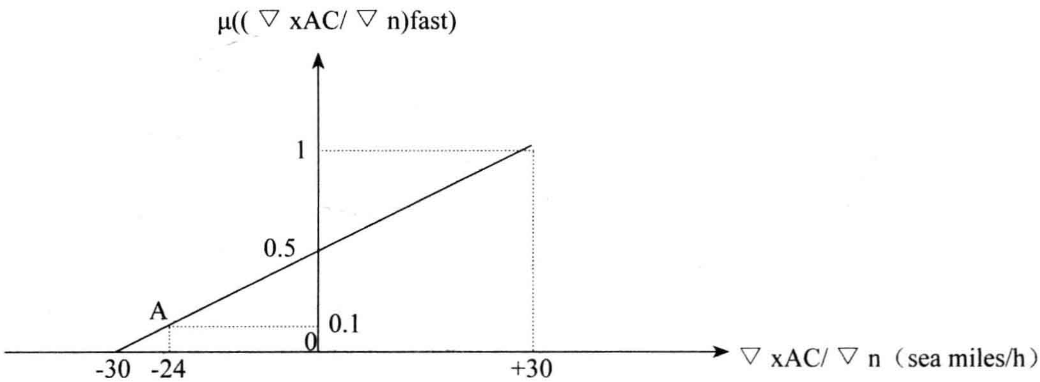


Fig. 39.6 Membership function for "($\nabla x_{AC}/\nabla n$) fast"

Suppose the scope of $\nabla x_W/\nabla n$ is $((\nabla x_W/\nabla n)_{min}, (\nabla x_W/\nabla n)_{max})=(-30$ meters/s, +30 meters/s), then the membership function for "$(\nabla x_W/\nabla n)$strong" is shown in Fig. 39.7.

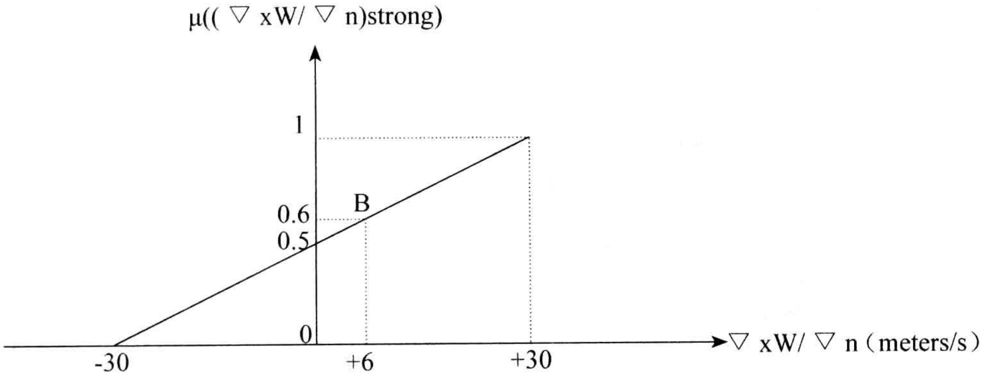


Fig. 39.7 Membership function for "$(\nabla x_W/\nabla n)$ strong"

Suppose the minimal deviation from the initial graduation of the x_M azimuth control of the missile is $-15°$, corresponding to $(\nabla x_M/\nabla n)_{min}$ of $(\nabla x_{AC}/\nabla n)_{min}$ and $(\nabla x_W/\nabla n)_{max}$; and the maximal deviation is $+15°$, corresponding to $(\nabla x_M/\nabla n)_{max}$ of $(\nabla x_{AC}/\nabla n)_{max}$ and $(\nabla x_W/\nabla n)_{min}$. Then the membership function of "deviation from the initial graduation of x_M azimuth control of the missile big" is shown in Fig. 39.8.

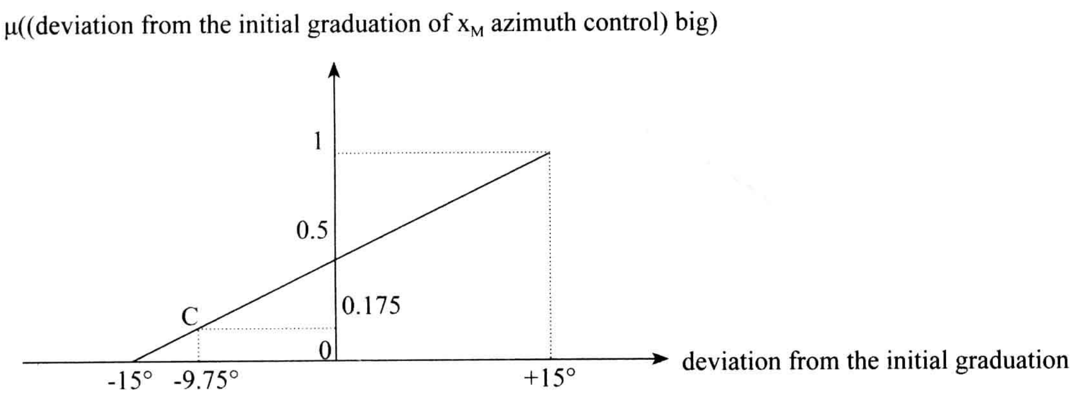


Fig. 39.8 Membership function for "(deviation from the initial graduation of x_M azi muth control of the missile) big"

Example 39.5: Suppose after the missile is launched, $\nabla x_{AC}/\nabla n = -24$ sea miles/h, which is converted to membership grade 0.1, shown in point A of Fig. 39.6; $\nabla x_W/\nabla n$ =+6 meters/s, which is converted to membership grade 0.6, shown in point B of Fig. 39.7; and the weight of $\nabla x_{AC}/\nabla n$ is 3, that of $\nabla x_W/\nabla n$ is 1. Then (39.49) becomes:

$((1 - 0.1)*3+0.6*1)/4 \leqslant_f^{-1} (1 -$ (deviation from the initial graduation of x_M azimuth control of the missile) big)

$0.825 \leqslant_f^{-1} (1 - 0.825)=0.175.$

Converting 0.175 back to the deviation from the initial graduation of x_M azimuth control of the missile, we obtain:

$30°*0.175 - 15° = -9.75°$

as shown in point C of Fig. 39.8. Therefore, the deviation from the initial graduation of x_M azimuth control of the missile should be adjusted to $-9.75°$.

References

Allen, James. 1995. Natural Languade Understanding (Second Edition). Pearson Education, Inc.

Bian, Jinian, Hongxi Xue, Ming Su, Weimin Wu. 2005. Digital Systems Design Automation, Beijing: Tsinghua University Press (Second Edition). (In Chinese)

Chen, Xiaoping. 2006. Introduction to Natural Deduction Logic. Guangzhou: Zhongshan University Press. (In Chinese)

Ding, Lijuan, Qiyuan Cheng. 2005. Numerical Computing Method (Second Edition). Beijing: Beijing Institute of Technology Press. (In Chinese)

Floyd, R.. 1967. "Assigning meaning to programs", Mathematical Aspect of Computer Science, XIX American Mathematical Society. 19-32.

Godel, Kurt. 1986. "On formally undecidable propositions of Principia Mathematica and related systems I", Collected Works Volume 1, 145-195. Oxford University Press, Oxford, Clarendon Press, New York.

Gordon, M. J. C., T. F. Melham. 1993. Introduction to HOL: A Theorem Proving Environment for Higher Order Logic. Cambridge University Press

Hamilton, A. G.. 1978. Logic for Mathematicians (Revised Edition). Cambridge University Press.

Hoare, C. A. R.. 1969. "An axiomatic basis for computer programming", Communications of the ACM, Vol. 12, Number 10, 576-580, 583.

Hu, Guangshu. 2005. Introduction to Digital Signal Processing. Beijing: Tsinghua University Press. (In Chinese)

Kleene, Stephen Cole. 1952. Introduction to metamathematics. North-Holland Publishing Co., Amsterdam.

Kneebone, G.T.. 2001. Mathematical Logic and the Foundations of Mathematics: An Introduction Survey. New York: Dover Publication, Inc.

Kolman, Bernard, Robert C. Busby, Sharon Ross. 1996. Discrete Mathematical Structures (Third Edition). Prentice-Hall Inc.

Kropf, Thomas. 1999. Introduction to Formal Hardware Verification. Springer-Verlag.

Li, Daogen. 2009. Modern Control Theory. Harbin: Harbin Institute of Technology Press.

Liu, Zhuanghu. 1993. Logical Calculi. Beijing: Chinese Academy of Social Sciences Press. (In Chinese).

Lloyd, J.M.. 1984. Foundations of Logic Programming. Springer-verlag.

Lu, JIanjiang, Yafei Zhang, Zilin Song. 2008. Studies and Applications of Fuzzy Association Rules. Beijing: Science Press. (In Chinese)

Mitchell, Tom M..1997. Machine Learning. McGraw-Hill Companies, Inc.

Morgan, Carroll. 1998. Programming from Specifications. Translation copyright 2002 by China Machine Press and CITIC Publishing House.

Pedrycz, Witold, Andrzej Skowron, Vladik Kreinovich. 2008. Handbook of Granular Computing. John Wiley&Sons, Ltd.

Rao, Jinghai, Peep Kungas, Mihhail Matskin. 2004. "Logic-based Web services composition: from services description to process model", Proceedings of the 2004 International Conference on Web Services, San Diego, USA, July 2004, IEEE.

Rich, E.. 1983. Artificial Intelligence. Singapore: McGraw-Hill.

Russell, S. J., P. Norvig. 2006. Artificial Intelligence: A Modern Approach (Second Edition). Beijing: Tsinghua University Press.

Shen, Fuxing. 1995. Introduction to Model Theory. Beijing: Beijing Normal University Press. (In Chinese).

Silberschatz, A., H. F. Korth, S. Sudarshan. 2002. Database System Concepts (Fourth Edition). New York: McGraw-Hill.

Suppes, Patrick. 1972. Axiomatic Set Theory. New York: Dover Publication Inc.

Takeuti, Gaisi. 1987. Proof Theory (Second Edition). New York: Elsevier Science Publishers B. V.

Varberg, Dale, Edwin J. Purcell, Steven E. Rigdon. 2000. Calculus (Eighth Edition). Prentice-Hall, Inc.

Winskel, Glynn. 1993. The Formal Semantics of Programming Languages: An Introduction. Massachusetts Institute of Technology.

Xia, Huosong. 2004. Data Warehouse and Date Mining Techniques. Beijing: Science Press. (In Chinese)

Yao, Tianshun, Jingpo Zhu, Li Zhang, Yin Yang. 2002. Natural Language Understanding—A Study that Make a Machine Understand Human Language (Second Edition). Beijing: Tsinghua University Press. (In Chinese).

Zhou, Xunwei. 2004. Mutually-Inversistic Mathematical Logic. Beijing: China Science and Technology Press. (In Chinese).

Zhou, Xunwei. 2009. Mutually Inverse Discrete Mathematics and its Applications. Beijing: Science Press. (In Chinese).

[General Information]
书名=互逆主义逻辑、数学和它们的应用=MUTUALLY-INVERSISTIC LOGIC, MATHEMATICS,AND THEIR APPLICATIONS
作者=周训伟著
页数=482
SS号=13436123
DX号=
出版日期=2013.03
出版社=中央编译出版社